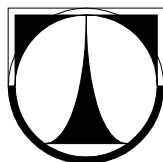


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií



DIPLOMOVÁ PRÁCE

Liberec 2009

Jiří Nový

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Využití Vision Development Module pro analýzu obrazu

Vision Development Module usage for image analysis

Diplomová práce

Autor:	Bc. Jiří Nový
Vedoucí práce:	Doc. Ing. Ivan Jaksch, CSc.
Konzultant práce:	Ing. Jiří Horčíčka

V Liberci 29. 5. 2009

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum:

Podpis:

Poděkování

Úvodem bych rád poděkoval vedoucímu diplomové práce Doc. Ing. Ivanu Jakschovi, CSc. za vedení práce, metodickou i odbornou pomoc a Ing. Jiřímu Horčíčkovi za konzultace v průběhu zpracování práce.

Abstrakt

Diplomová práce v úvodu seznamuje s programovým balíkem NI Vision. Soustředí se především na NI Vision Assistant a NI LabWindows/CVI. Je zde popsáno prostředí, práce v něm a asi nejdůležitější část tohoto programového balíku, což jsou knihovny pro zpracování obrazu. Na začátku je také předvedeno několik běžných funkcí zpracování obrazu.

Další část je věnována představení praktické úlohy, hardwarových a softwarových prostředků a testování funkcí. Konkrétně se v praktické úloze jedná o kontrolu kvality stírání skla automobilu, která je vyvíjena pro firmu Škoda auto a.s., kde byla doposud kvalita stírání plochy skla automobilu hodnocena subjektivně jejich zaměstnancem. Program vyvíjený v této práci by měl objektivně hodnotit kvalitu stírání skla.

V závěru je zhodnocen jak softwarový balík NI Vision, tak program v něm vyvíjený.

Autor diplomové práce:	Bc. Jiří Nový
Vedoucí diplomové práce:	Doc. Ing. Ivan Jaksch, Csc.
Konzultant diplomové práce:	Ing. Jiří Horčíčka
Název diplomové práce:	Využití Vision Development Module pro analýzu obrazu

Klíčová slova

Využití Vision Development Module, analýza obrazu, Houghova transformace

Abstract

Master's thesis in the introduction acquaints readers with the software package NI Vision .It focuses primarily on the NI Vision Assistant and NI LabWindows / CVI. It describes the background here, work there. Libraries are probably the most important part of this package for image processing. At the beginning it presented several common image processing functions.

Another part is devoted to the presentation of practical exercise, hardware and software and testing functions. Specifically, it is the practical role of the quality control of glass blurring car. It is developed for the company Škoda auto a.s.. So far, the quality of the glass surface blurring car subjectively assessed by their employee. The program developed in this work should objectively evaluate the quality of glass blurring.

In conclusion, it is evaluated how the NI Vision software package, and program it developed.

Author of the thesis:	Bc. Jiří Nový
Head of the thesis:	Doc. Ing. Ivan Jaksch, Csc.
Consultant of the thesis:	Ing. Jiří Horčíčka
Theme of the thesis:	Vision Development module usage for image analysis

Keywords

Usage Vision Development Module, Image Analysis, Hough Transform

OBSAH

1 Úvod.....	11
2 NI Vision Development Module.....	12
2.1 Algoritmy.....	12
2.1.1 Zpřesnění obrazu.....	12
2.1.2 Kontrola přítomnosti.....	13
2.1.3 Hledání charakteristických rysů.....	13
2.1.4 Měřicí funkce.....	14
2.1.5 Identifikace částí.....	14
2.1.6 Výkony algoritmů.....	15
2.1.7 Integrace s jinými prostředky.....	16
2.2 NI Vision Assistant.....	16
2.2.1 Ovládání.....	17
2.2.2 Funkce NI Vision Assistant.....	18
2.2.2.1 Obrazové funkce.....	19
2.2.2.2 Barevné funkce.....	20
2.2.2.3 Šedotónové funkce.....	21
2.2.2.4 Binární funkce.....	22
2.2.2.5 Funkce počítačového vidění.....	22
2.2.2.6 Funkce identifikační.....	23
2.3 LabVIEW.....	24
2.4 LabWindows/CVI.....	25
2.4.1 Prostředí.....	25
2.4.2 Menu.....	26
2.4.3 Funkce NI Vision.....	28
2.4.3.1 NI Vision.....	31
2.4.4 Vytvoření programu.....	33
3 Hodnocení kvality stírání u zkušebního okenního skla.....	34
3.1 Zadání úlohy.....	34
3.1.1 Norma.....	35
3.1.2 Zkušební okenní sklo.....	36
3.2 Hardwarové prostředky.....	36
3.2.1 Kamery.....	36
3.2.2 Objektivy.....	39
3.2.3 Osvětlení.....	40
3.2.4 Konstrukce.....	43
3.3 Aplikace Hodnocení kvality stírání u zkušebního okenního skla.....	44
3.3.1 Testované metody.....	44
3.3.1.1 Prahování.....	47
3.3.1.2 Houghova transformace.....	51
3.3.1.3 Průměrná hodnota jasu v oblasti.....	53

3.3.1.4 Diferenční metoda.....	56
3.3.2 Program.....	57
3.3.2.1 Zdrojový kód.....	58
3.3.2.2 Grafické uživatelské rozhraní.....	62
4 Závěr.....	65
Seznam literatury.....	66
Přílohy:.....	67

SEZNAM ILUSTRACÍ

Obr. 2.1: Měření vzdálenosti, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: < www.ni.com >.....	12
Obr. 2.2: Kontrola přítomnosti, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: < www.ni.com >.....	13
Obr. 2.3: Hledání charakteristických rysů, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: < www.ni.com >.....	13
Obr. 2.4: Ukázka výpočtu počtu buněk, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: < www.ni.com >.....	14
Obr. 2.5: Identifikace částí, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: < www.ni.com >.....	14
Obr. 2.6: NI Vision Assistant 8.6.0.....	17
Obr. 2.7: Funkce: Obrazové.....	19
Obr. 2.8: LabVIEW.....	24
Obr. 2.9: LabWindows/CVI.....	26
Obr. 2.10: Výběr vzhledu LabWindows/CVI.....	27
Obr. 3.1: Zkušební okenní sklo.....	36
Obr. 3.2: Basler scA1600-FM.....	37
Obr. 3.3: Snímání 4-mi kamerami.....	38
Obr. 3.4: Objektiv.....	39
Obr. 3.5: Zobrazovací rovina.....	40
Obr. 3.6: Vypnuté ELP-A4 a zdroj.....	41
Obr. 3.7: Zapnuté ELP-A4, stěrač s gumičkou a zdroj.....	41
Obr. 3.8: Napájení EL panelů.....	42
Obr. 3.9: Struktura EL panelů.....	42
Obr. 3.10: Osvětlení pomocí EL panelů.....	43
Obr. 3.11: Line Profil 1.....	46
Obr. 3.12: Line Profil 2.....	46
Obr. 3.13: Manuální prahování, světlé objekty.....	48
Obr. 3.14: Manuální prahování, šedé objekty.....	49
Obr. 3.15: Manuální prahování, tmavé objekty.....	49

Obr. 3.16:Doba vykonávání prahování.....	51
Obr. 3.17: Parametricky popsaná úsečka.....	52
Obr. 3.18: Okno programu ve kterém se nastavovali ROI.....	54
Obr. 3.19: Část okna s nalezenými oblastmi.....	55
Obr. 3.20: Diference, od obrazu s chybou se odečte obraz bez chyby a výsledkem je hledaná chyba (zvýrazněná pomocí prahování).....	56
Obr. 3.21:Doba vykonávání prahování.....	57
Obr. 3.22: Funkce která načítá obraz z kamer do obrazových polí	59
Obr. 3.23: Funkce zobraz.....	60
Obr. 3.24: Funkce Diference.....	60
Obr. 3.25: Funkce Prum.....	61
Obr. 3.26: Funkce pomocné.....	62
Obr. 3.27: Okno grafického uživatelského rozhraní programu.....	63
Obr. 1: Lokální prahování, Niblack (světlé objekty, tmavé objekty).....	68
Obr. 2: Lokální prahování, Niblack (světlé objekty, tmavé objekty).....	69
Obr. 3: Lokální prahování, Background Correction (světlé objekty, tmavé objekty).....	69
Obr. 4: Automatické prahování, Clustering (světlé objekty, tmavé objekty).....	69
Obr. 5: Automatické prahování, Entropy (světlé objekty, tmavé objekty).....	70
Obr. 6: Automatické prahování, Metric (světlé objekty, tmavé objekty).....	70
Obr. 7: Moments, Clustering (světlé objekty, tmavé objekty).....	70
Obr. 8: Inter Variance, Clustering (světlé objekty, tmavé objekty).....	71
Obr. 9: Strana 19 z normy TL 934.....	72
Obr. 10: Strana 20 z normy TL 934.....	73
Obr. 11: Strana 21 z normy TL 934.....	74
Obr. 12: Vytváření ROI u funkce průměrování.....	75
Obr. 13: Nalezené ROI s chybou větší než odchylka	75

Seznam tabulek

Tabulka 1: Porovnání rychlostí zpracování, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-06]. Dostupný z WWW: < www.ni.com >.....	15
Tabulka 2: Přesnost kamery.....	38
Tabulka 3: Objektivy.....	39

1 Úvod

Počítačové vidění se v principu od toho lidského neliší. Světlo dopadající na přijímač je zpracováno a zobrazeno. Rozdíly jsou ovšem ve způsobech jakým je obraz přijímán, zpracován a zobrazen.

Přijímačem u člověka je lidské oko, které je schopno rozeznat asi 400 000 barevných odstínů a 60 odstínů šedi. Je omezeno světelnými podmínkami, vzdáleností a velikostí objektu. Nedokonalosti lidského oka se můžou za pomoci různých technických prostředků zdokonalit (dalekohledem, brýlemi nočního vidění, atd.). Dalším omezením je stáří či únava člověka, od čehož se pomalu dostáváme ke zpracování, které je u člověka dáno vývojem. Vnímání tvarů a barev je ovlivněno také psychikou a dosavadními zkušenostmi osobnosti. Díky těmto zkušenostem může při zpracování dojít k optickým klamům. Zobrazení je jedinečné a člověk vidí jen svůj zpracovaný obraz, který je jeho dalším vjemem. Svůj viditelný obraz může jen popsat.

Přijímačem u počítačů a různých strojů je kamera. Její kvalita je omezena jak technickým pokrokem, tak cenou. Důležitým faktorem bývá také nasvícení. Na rozdíl od biologického zpracování obrazu člověkem je obraz zpracováváný počítačem výsledkem programování a čím lepší obraz snímáme, tím je jednodušší naprogramování aplikace. Právě zpracováním obrazu se bude zabírat tato diplomová práce. Ve zpracování je největší rozdíl mezi lidským a počítačovým viděním. Výsledek zpracování obrazu poté buďto vyhodnotí počítač, nebo se zobrazí člověku.

Výhodou počítačového zpracování obrazu je přesnost, nenáchylnost k přehlédnutí, rychlé zpracování mnoha různých příkazů a neúnavný běh. Největší nevýhodou je naprogramování onoho zpracování obrazu. Počítači se nedají předat zkušenosti člověka a zpracování se musí naprogramovat. Počítač nevnímá obraz jako obraz, ale jako data, která může různými způsoby zpracovávat a vyhodnocovat.

2 NI Vision Development Module

NI Vision Development module je produkt Firmy National Instruments. Tato firma již více jak deset let udává směr v oblasti počítačového vidění. Jedná se o softwarový balík, který obsahuje mnoho výkonných funkcí určených ke zpracování obrazu, pro více programovacích prostředí, jako je NI LabVIEW, C++ , Visual Basic, a .NET. Navíc je kompatibilní s velkým množstvím hardware určeného ke zpracování obrazu (tisíce kamer, stovky OCR zařízení,...).

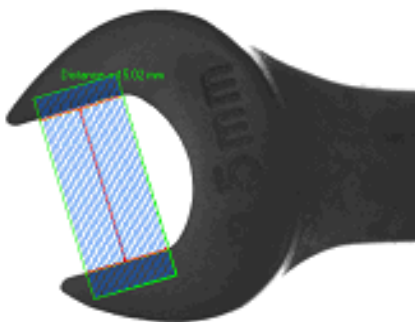
2.1 Algoritmy

Je velmi důležité, aby softwarový nástroj správně a přesně měřil důležité části, nebo vlastnosti objektu. Pokud by tento software nebyl přesný a spolehlivý, nebyla by nám k ničemu sebelepší kamera ani výkonný počítač.

NI Vision Development Module obsahuje stovky přesných a spolehlivých funkcí. Pět nejběžnějších funkcí počítačového vidění je popsáno níže.

2.1.1 Zpřesnění obrazu

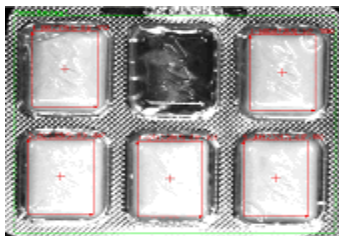
Ke zvýraznění hran, nebo k vytažení důležitých informací se používají filtrační nástroje. K odstranění nelineárních a pohledových chyb způsobených zkreslením objektivu a umístěním kamer se používají kalibrační nástroje. Kalibrační nástroje můžeme také použít k získání skutečných jednotek měření (obr. 2.1). Aplikace tudíž mohou vracet hodnoty v jednotkách (mikronech, milimetrech,...) namísto v pixelech.



Obr. 2.1: Měření vzdálenosti, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6].
Dostupný z WWW: <www.ni.com>.

2.1.2 Kontrola přítomnosti

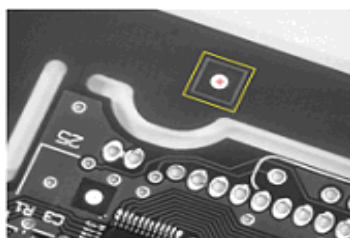
Jedná se o nejjednodušší inspekční funkci počítačového vidění. Ke kontrole přítomnosti nějaké části, či celku se může použít některá z barev (načte se vzorová barva, jako u příkladu na obr 2.2), nebo se použije histogram.



Obr. 2.2: Kontrola přítomnosti, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: <www.ni.com>.

2.1.3 Hledání charakteristických rysů

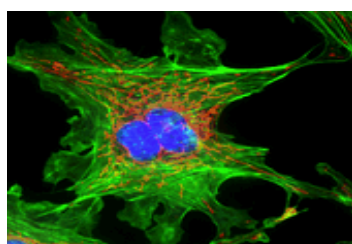
Hledání charakteristických rysů (obr 2.3) je důležité při ladění, nebo určení přesného umístění objektu, které slouží jako standart pro všechny následující kontroly. Funkce, které je možno použít k nalezení shodných rysů jsou, detekce hran, šedotónové škálování odpovídající vzoru, porovnání tvaru, porovnání geometrických obrazců, porovnání a hledání barevné shody se vzorem. Nástroj vrátí pozici objektu (X, Y) a úhel otočení s přesností až na jednu desetinu pixelu. Geometrická shoda je imunní vůči překrývajícím se objektům, nebo objektům, které změnili své měřítko.



Obr. 2.3: Hledání charakteristických rysů, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: <www.ni.com>.

2.1.4 Měřicí funkce

Jedním z nejčastějších důvodů k použití počítačového vidění je měření (obr 2.4). Obvykle se používá detekce hran, granulometrie geometrické funkce a nástroje pro měření vzdálenosti, průměru, celkového počtu, úhlu a plochy. Ať už jde o výpočet celkového počtu buněk pod mikroskopem, nebo výpočet úhlu na nějaké součástce, tak tyto nástroje vrátí vždy hodnoty zjištěné v dané oblasti, či hodnotu *fail/pass* (neúspěšná/úspěšná).



Obr. 2.4: Ukázka výpočtu počtu buněk, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: <www.ni.com>.

2.1.5 Identifikace částí

Identifikace části (obr. 2.5) je důležitá pro zjištění shody, sledování a ověřování. Přímé identifikační metody obsahují čtení čárového kódu, nebo datového kódu jako je maticový kód a PDF 417. Novější metody používají takzvané učící se OCR, nebo strojové třídění. Výstupem identifikační části bývá nejčastěji text, nebo řetězec, na rozdíl od výsledků různých měření, či určování úspěšnosti a neúspěšnosti.



Obr. 2.5: Identifikace částí, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-6]. Dostupný z WWW: <www.ni.com>.

2.1.6 Výkony algoritmů

Dvěma nejdůležitějšími faktory při výběru systému počítačového vidění často bývá přesnost a snadnost použití. Dalším důležitým faktorem je rychlost výpočtu. Nezáleží na tom, kolik stovek algoritmů je na výběr, nebo jak rychle s nimi lze tvořit aplikace, pokud jsou totiž jednotlivé kontrolní aplikace neefektivní a trvají příliš dlouho, tak jsou často k ničemu.

NI Vision je vysoce optimalizovaný software s velkým množstvím patentovaných algoritmů. Výsledkem je, že tento softwarový balík je v oblasti počítačového vidění nejrychlejší na světě. Ve srovnání s ostatními předními poskytovateli softwaru počítačového vidění, je NI Vision software rychlejší, jak je znázorněno v tabulce 1.

	NI Vision Software, rychlost (ms)	Předními poskytovatele softwaru k počítačovému vidění, rychlost (ms)	Násobek vyšší rychlosti NI
Histogram	0,91	2,03	2.2X
Geometrická transformace	3,1	10,3	3.3X
Morfologie	1,8	5,9	3.3X
OCR	3,3	5,9	1.8X
Geometrická Shoda	93,0	149,8	1.6X
Klasifikace objektu	7,5	—	—

Tabulka 1: Porovnání rychlostí zpracování, zdroj www.ni.com [on line]. 2007 [cit. 2009-04-06]. Dostupný z WWW: <www.ni.com>.

2.1.7 Integrace s jinými prostředky

Počítačové vidění často bývá součástí mnohem větších kontrolních systémů. V průmyslové automatizaci může být aplikace počítačového vidění použita k výstupní kontrole správného seřazení produktů; sdělení výsledků řadiči robota, PLC, či automatickému programovatelnému regulátoru; uložení obrázků a dat do sítě, na server, nebo sdílení parametrů inspekce a výsledků na lokální, či vzdálené uživatelské rozhraní.

Firma National Instruments nevyrábí zařízení jako jsou osvětlení, kamery, nebo optiky, místo toho úzce spolupracuje s těmi, kdo se výrobou těchto zařízení zabývají. Díky tomu podporuje velké množství těchto zařízení. Na webových stránkách NI Vision Product Partner se nacházejí informace a rady k potřebným komponentám, či informace o tom, kde tyto komponenty pořídit, či objednat.

2.2 NI Vision Assistant

Jedním z úkolů je seznámení s NI Vision Assistant, jenž je součástí Vision Development Module. Konkrétně budu popisovat Vision Assistant verze 8.6.0. Jedná se o program, který usnadňuje vývoj aplikací počítačového zpracování obrazu, který je proto vhodný hlavně pro předzpracování obrazu, ale dají se v něm vytvořit i jednodušší skripty, které se poté přepíší, nebo vyexportují do nějaké jiné aplikace.

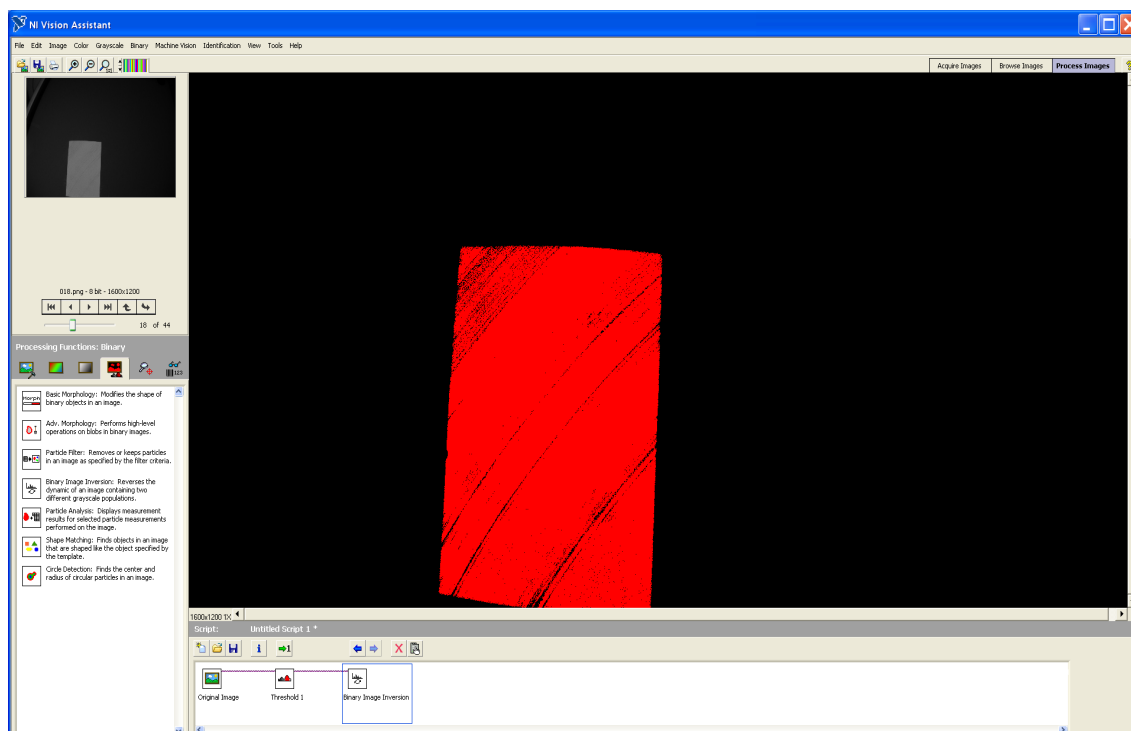
NI Vision Assistant je GUI¹ prostředí, do kterého se načítají obrázky, se kterými pracuje. Hledají se nejvhodnější funkce, které nám pomáhají najít požadovaný výsledek. Díky WYSIWYG² je výsledek změn ihned zobrazen na obrazovce. Jak je vidět na obrázku 2.6, okno programu je rozděleno na několik částí.

V horní liště se nachází klasické menu a pod ním jsou tlačítka pro práci s obrazem. Níže je obrazovka rozdělena na 4 části. V levé horní části se zobrazují

1 GUI = grafické uživatelské rozhraní

2 WYSIWYG umožňuje uživateli zobrazovat výsledek práce už ve fázi úprav či vytváření.

všechny načtené obrázky, mezi kterými se mohou přepínat, a které lze zobrazit pomocí tlačítka v pravé horní části. Zde je poté vidět, co ta či ona funkce s obrázkem udělá a zda-li to dělá tak, jak si to autor představuje. Funkce se nacházejí v levé dolní části. Tyto funkce jdou skládat za sebe a pomocí jejich kombinací lze dosáhnout potřebných výsledků. Jak jsou tyto funkce poskládány za sebou je vidět v pravé dolní části na obr. 2.6. Z celého řetězce vzniká skript, který jde nejen ukládat a načítat. Ze skriptu je možné vygenerování kódu v jazyce C, či přímo do programu NI LabVIEW.





Obr. 2.6: NI Vision Assistant 8.6.0

2.2.1 Ovládání

Základním ovládacím prvkem je Menu. Obsahuje 11 hlavních položek pro práci se soubory, nastavení, úpravu obrazu, nápovědu a tak podobně. Nejvyužívanější je panel funkcí sloužící ke zpracování obrazu a panel pro práci s obrazy.

První položkou menu je File, která slouží k práci se soubory. Lze zde načítat a ukládat skripty, procházet, načítat, ukládat a tisknout obrázky. Je zde i základní nastavení programu a jeho ukončení. Dalšími položkami jsou Image, Color, Grayscale,

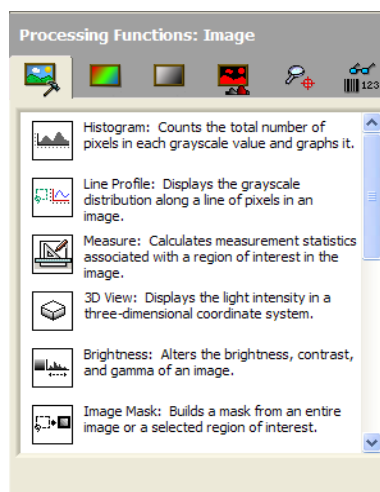
Binary, Machine Vision a Identification. Tyto položky jsou identické s panelem funkcí pro zpracování obrazu a jsou podrobněji popsány níže, v kapitole 2.2.3. Předposlední je položka Tools sloužící k nastavení, doladění a exportování skriptu do kódu programovacího jazyka C, či do LabVIEW Vision. Nachází se zde také *Performance Meter*, který slouží k zjištění časové náročnosti skriptu a jeho jednotlivých částí. Zobrazuje se zde jak průměrný, tak i nejlepší čas, nejhorší čas a časová odchylka. Poslední položkou menu je Help. Obsahuje kontextovou nápovědu a Online nápovědu, a příklady pro různé typy úloh. Jsou zde i základní informace o programu a patentech.

Panel pro práci s obrazy obsahuje ikony pro základní práci s obrázky, jako je načtení, uložení, procházení, listování mezi nimi, přibližování, oddalování, či tisk. Pomocí ikony šipky vpravo dolů  překopírujeme obraz z panelu pro práci s obrazy do okna, kde se obraz za pomoci funkcí zpracovává. Zpracovaný obraz můžeme načíst zpět do panelu pro práci s obrazy pomocí šipky doleva nahoru  .

2.2.2 Funkce NI Vision Assistant

Funkce jsou rozděleny do šesti základních skupin:

- Obrazová
- Barevná
- Šedotónová
- Binární
- Počítačového vidění
- Identifikační



Obr. 2.7: Funkce: Obrazové

Tyto funkce zobrazuje 6 ikon v horním řádku, jak je vidět na obrázku 2.7. Pod těmito ikonami jsou pod sebou umístěny funkce, které náleží k právě označené základní skupině. Tyto jednotlivé funkce jsou taktéž označeny ikonou a navíc stručným popisem.

2.2.2.1 Obrazové funkce

- **Histogram**: Počet pixelů každé šedotónové hodnoty zobrazený grafem a vypsáním informací. Výsledky je možno uložit, či exportovat do Excelu. Lze přepínat mezi dvěma mapovacími módy, buď Lineárním a nebo logaritmickým.
- **Line profil**: Zobrazuje odstíny šedi pixelů, umístěné podél linie, která je natažena na obrazu.
- **Measure**: Spočítá míry statisticky spojené s oblastí, která nás zajímá na obrazu. Lze vybrat, zda-li chceme měřit pozici, vzdálenost, úhel, či plochu.
- **3D View**: Zobrazuje intenzitu světla ve tří-dimenzionálním souřadnicovém systému. Výsledný obraz pak můžeme natáčet, zvyšovat výšku, či invertovat.
- **Brightness**: Mění jas, kontrast a gama na obrazu.

- Set Coordinate System: Sestaví souřadnicovým systémem založeným na umístění a orientaci referenční funkce.
- Image Mask: Sestaví masku celého obrazu, nebo pouze z oblasti obrazu, která nás zajímá.
- Geometry: Upravuje geometrické reprezentace obrazu.
- Image Buffer: Ukládá obraz do zásobníku, ze kterého je může poté přechít (lze mít uloženo až deset obrazů).
- Get Image: Otevře nový obraz ze souboru.
- Image Calibration: Kalibrace obrazu k převedení naměřených hodnot na skutečné hodnoty.
- Calibration from Image: Platí kalibrační hodnoty uložené v obrazu, které se použijí na aktuální obraz.
- Image Correction: Transformuje zkreslený obraz podle kalibračního nastavení na opravený obraz.
- Overlay: Překrývá kontury, rastry, nebo text na obrazu, bez poškození základních obrazových dat.
- Run LabVIEW VI: Provozuje uživatel. (viz. kapitola 2.3)

2.2.2.2 Barevné funkce:

- Color Operators: Zvládají aritmetické a logické operace s obrazem.
- Color Plane Extraction: Vyjme 3 barevné roviny (RGB, HSV, nebo HSL) z obrazu.
- Color Threshold: Používá se prahování obrazu ve třech barevných rovinách a výsledek se zobrazí jako 8-bitový obraz.
- Color Location: Vyhledává barvy v barevném obrazu

- Color Pattern Matching: Kontroluje přítomnost šablony v celém obrazu, nebo jen v oblasti, která nás zajímá.
- Color Matching: Naučí se barevný obsah regionu obrázku a porovnává to s barevným obsahem jiného regionu.

2.2.2.3 Šedotónové funkce:

- Lookup Table: Zvyšuje kontrast a jas pomocí aplikace převodní tabulky použité na obraz. Jedná se o tabulky: vyrovnání, opačné, logaritmické, exponenciální, kvadratické, odmocninové, power X a power 1/X (u power X a power 1/X jde nastavit hodnota power).
- Filters: Připravují obraz na zpracování, takže lze extrahovat pouze informace, které z obrazu potřebujeme. Tato funkce obsahuje filtry vyhlazovací (s nízkou propustností, místní průměr, Gaussův a medián), filtry hranové (Laplasův, diferenciální, vyjasňovací, Sobel, Roberts) a konvulační filtry (zvýraznění detailů a výběrový).
- Grey Morphology: Upravuje tvar objektů obrazu (dilatace, eroze, uzavření, otevření, řádné uzavření, řádné otevření a automatický medián).
- FFT Filter: Používá frekvenční filtr na obraz. Obsahuje 2 funkce, seřiznutí a ztenčení
- Threshold: Na výběr jsou tyto typy prahování: manuální prahování, lokální prahování Niblack, lokálního prahování - korekce pozadí, automatické prahování – shlukování, automatické prahování entropické, automatické prahování metrické, automatické prahování momentové, automatické prahování – meziodchylka.
- Watershed Segmentation: Provede segmentaci na základě rozvodí.
- Operators: Provádí aritmetické a logické operace nad obrazem.
- Conversion: Převádí současný typ obrazu na jiný typ obrazu (8-mi bitový, 16-ti bitový, Float).
- Quantify: Kvantifikuje obsah obrazu nebo oblast, kterou vybereme.
- Centroid: Spočítá energii středu z obrazu

2.2.2.4 Binární funkce:

- Basic Morphology: Upravuje tvar objektů binárního obrazu a obsahuje tyto funkce: erozi objektu, dilataci objektu, otevření objektu, uzavření objektu, řádné otevření, řádné uzavření, vnitřní gradient, vnější gradient, automatický medián, hustota, řídkost.
- Advance Morphology: Zvládne vysokoúrovňové operace na binárním obrazu, konkrétně tyto: odstranit malé objekty, odstranit velké objekty, odstranit hraniční objekty, vyplnit díry, konvexní obal, kostra, oddělení objektů, označení objektů, vzdálenost, Danielsson, segmentace obrazu.
- Particle Filter: Odebere, nebo ponechá částice, specifikované podle kritérií filtru.
- Binary Image Inversion: Prohodí binární hodnotu každého pixelu obrazu.
- Particle Analysis: Zobrazuje výsledky měření jednotlivých částic celého obrazu.
- Shape Matching: Vyhledává objekt na obrazu, který je porovnáván s objektem specifikovaným v šabloně.
- Circle Detection: Najde střed a poloměr kruhové částice na obrazu.

2.2.2.5 Funkce počítačového vidění:

- Edge Detector: Detekuje hrany na šedotónovém obrazu.
- Straight Edge (Rake): Vyhledává přímou hranu v oblasti, která nás zajímá.
- Advance Straight Edge: Vyhledává přímou hranu v oblasti, která nás zajímá.
- Circle Edge (Spoke): Vyhledává kruhové hrany v oblasti, která nás zajímá.
- Clamp: Měří vzdálenosti dělicí hrany objektu.
- Pattern Matching: Kontroluje přítomnost šablony v celém obrazu, nebo jen v oblasti, která nás zajímá, na základě intenzity.

- Geometric Matching: Kontroluje přítomnost šablony v celém obraze, nebo jen v oblasti, která nás zajímá, na základě geometrie.
- Shape Detection: Hledá geometrické tvary v obraze, nebo pouze v oblasti, která nás zajímá.
- Golden Template Comparsion: Porovnává oblasti obrázku k naučení se šablony a vrací nalezené rozdíly obrazu.
- Caliper: Zobrazuje výsledky měření prováděné na vybraných bodech.

2.2.2.6 Funkce identifikační:

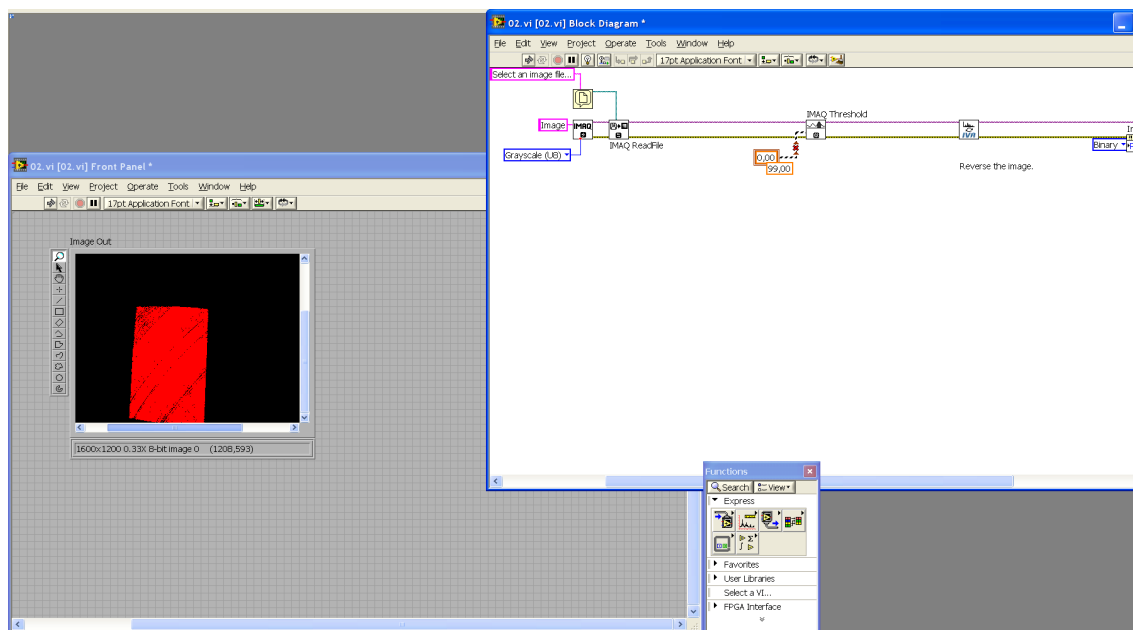
- OCR / OCV: Čtení znaků v oblasti na obraze.
- Clasification: Klasifikace částic v oblasti na obraze.
- Barcode Reader: Čtení 1D čárového kódu.
- Data Matrix Reader: Čtení maticového kódu (2D čárového kódu).
- QR Code Reader: Čtení QR kódu.
- PDF417 Code Reader: Čtení PDF417 Kódu.

2.3 LabVIEW

LabVIEW je další částí programového balíku NI Vision a proto zde bude krátce představen. Jedná se o LabVIEW verze 8.6.0. Programování probíhá za pomoci stavového diagramu.

Jak je vidět na obrázku 2.8, prostředí se standardně skládá z několika oken. Zde konkrétně slouží levé okno k vytvoření grafického uživatelského rozhraní programu, tzv. GUI. Zde se mohou umisťovat různé ovládací prvky (tlačítka, přepínače, ...), zobrazovací prvky (diody, grafy, indikátory, ...) a jiné (textová pole, tabulky, obrazy,...). Dále je zde velké množství funkcí pro zpracování obrazu, které jsou popsány v kapitole 2.4.3.

V pravém okně na obrázku 1.8 lze vidět stavový diagram, pomocí něhož se zde programuje. Různým přetahováním myši lze jednotlivé objekty upravovat a vytvářet objekty nové. Spojovat je lze dle potřeb. Poklepání na objekt, který chceme nějak nastavit se zobrazí další okno (jako např. malé okénko dole na obrázku 2.8). V tomto okně se provádějí potřebná nastavení.



Obr. 2.8: LabVIEW

2.4 LabWindows/CVI

LabWindows/CVI³ je událostmi řízené ANSI C vývojové prostředí od firmy National Instruments. Původně byl program v roce 1987 vytvořen pro DOS, ale brzy byl předělán (a přejmenován) pro platformu Microsoft Windows, nyní je označován jako CVI. LabWindows/CVI používá ty samé knihovny a ty samé moduly, jako výše popsany NI LabVIEW (kapitola 2.3) a díky tomu je s ním silně kompatibilní. Na rozdíl od LabVIEW, které je určeno spíše pro vědce a odborníky, je LabWindows/CVI určeno především softwarovým inženýrům, pro které bývá jednodušší textový jazyk (jazyk C), než stavový diagram.

Zde popisovaný program je verze 9.0.0. Systémové požadavky jsou dle výrobce: PC Pentium 1GHz nebo vyšší, Microsoft Windows Vista/XP, nebo Windows 2000, alespoň se Service Pack 3, rozlišení 1024 x 768 (nebo vyšší), minimálně 128 MB RAM (256MB RAM doporučeno), 200 MB volného místa na pevném disku a myš.

2.4.1 Prostředí

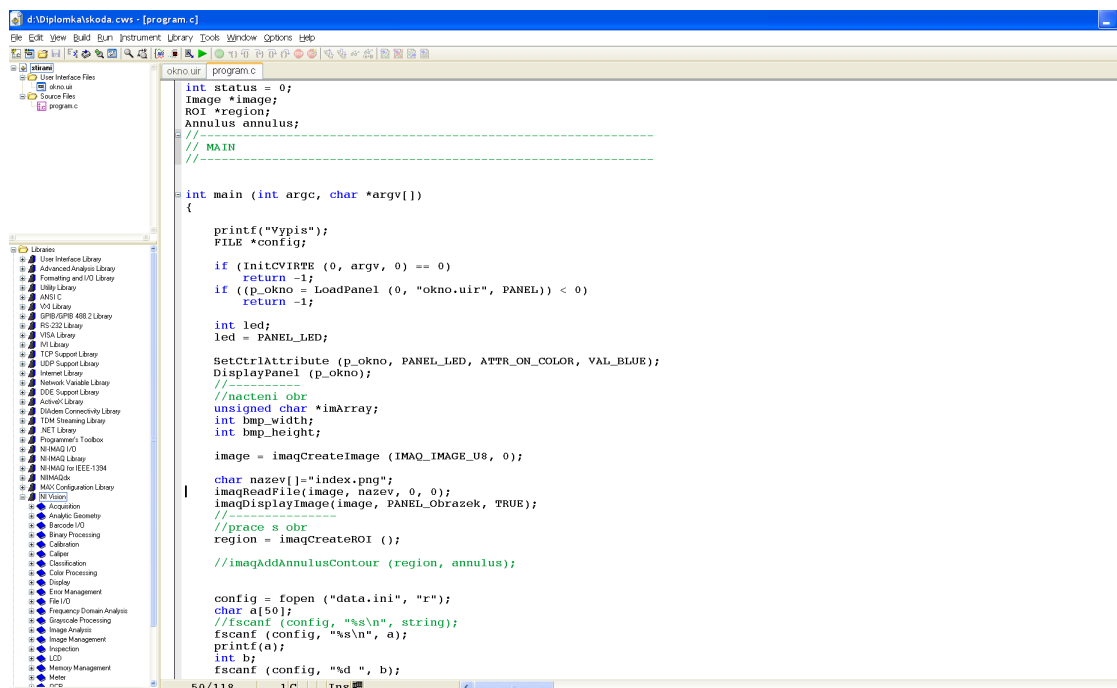
Jedná se o editor programovacího jazyka C se Syntax Highlighting⁴, které lze samozřejmě vypnout. Na obrázku 2.9 je vidět okno spuštěného editoru⁵. V horní části se nachází menu, které je popsáno v následující kapitole (2.4.2). Pod menu se nacházejí ikony usnadňující a urychlující ovládání. Ikonky je samozřejmě možné přidávat a odebírat. Lze také nastavit klávesové zkratky pro rychlejší ovládání.

Okno je standardně rozděleno na pravou a levou část. V pravé části je editor, který může být jak grafický, tak textový. Typ editoru závisí na tom, jestli se vytváří ovládací okno uživatelského rozhraní, či píše nebo ladí zdrojový kód. Levá část je rozdělena na horní a spodní. V horní části je vidět stromová struktura projektu. Ve spodní je vidět stromová struktura knihoven a připojených zařízení. Toto zobrazení lze poté nastavit pomocí Menu View.

3 CVI = C for Virtual Instrumentation.

4 **Syntax highlighting** je vlastnost některých editorů, které zobrazí text v různých barvách a fontech podle kategorie podmínky.

5 Jedná se o výchozí vzhled ve zobrazení integrated view



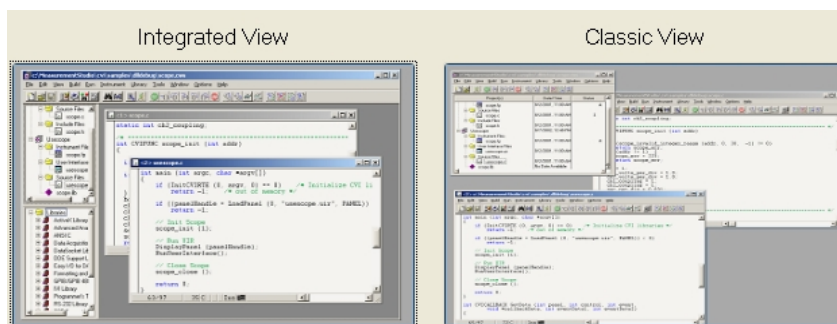
Obr. 2.9: LabWindows/CVI

2.4.2 Menu

Menu obsahuje 11 hlavních položek. První hlavní položka se jmenuje File (Soubor). Ta obsahuje základní funkce pro práci se soubory, jako je vytváření, načítání, či ukládání.

Další hlavní položkou je Edit, kde se jak název napovídá upravuje pracovní oblast, či projekt. Zde je také možno přidávat soubory různých typů do projektu, nebo je vyhledávat.

Následující hlavní položkou je View. Zde je možnost výběru dvou typů zobrazení (jak je vidět na obrázku 2.10), a to integrated view, kde jsou všechna okna integrována v jednom okně, nebo Classic View, kde je každý soubor ve svém vlastním okně. Dále zde lze vypnout, či zapnout zobrazení různých oken a listů.



Obr. 2.10: Výběr vzhledu LabWindows/CVI

Build obsahuje příkazy ke kompilaci, nastavení kompilace, ladění, vkládání hlaviček, k vybudování a propojování projektů, označování projektů pro kompilaci a vytváření aplikačních souborů.

Hlavní položka Run obsahuje příkazy ke spuštění a ladění programu. Dají se zde nastavit breakpointy⁶, sledovat výrazy, dělat kroky v kódu podle podmínek, zobrazování hodnot a další.

Instrument je dynamické menu. Toto menu obsahuje seznam nahrávaných ovladačů zařízení a příkazy k načtení, vyjmutí a editaci zařízení. Pokud se nějaké zařízení načtete, přidá se jeho jméno do seznamu a pokud ho odpojíte, ze seznamu se odebere. Jestliže zvolíte jméno nějakého zařízení v tomto menu, můžete vstoupit do jeho funkčního panelu⁷. Ovladače zařízení se objevují také pod stromovou strukturou knihoven.

Library menu slouží k přístupu k funkčnímu panelu knihoven LabWindows/CVI. Spouštějí se zde knihovní funkce a vkládají se do námi zvoleného okna.

Tools menu, slouží k nastavení a úpravě programu. Lze zde vytvořit ActiveX řadič, vytvořit a upravovat ActiveX server, vytvořit .NET řadič, pracovat s různými zařízeními na zpracování obrazu, vytvořit Visual Studio projekt, pracovat s tabulkovým procesorem Excel, či třeba vytvořit ikonu výslednému programu.

⁶ **Breakpoint** usnadňuje programátorovi ladění programu, program se zastaví na breakpointu a čeká na akci programátora.

⁷ Jedná se o dynamické menu, obsahující seznam ovladačů, příkazů k jejich nahrávání a editační software.

Následující položka menu Windows obsahuje příkazy k úpravě vzhledu oken, zobrazování oken a zobrazování stavových řádků. Zde je možnost například nastavit zobrazení více oken se zdrojovým kódem nad sebe, či vedle sebe, srovnat je za sebou, vyjmout je z okna programu, či všechny minimalizovat, nebo zavřít. Také je zde možné zobrazit a skrýt různá okna, výstupní, chybová, hodnot proměnných atp.

Předposlední je Options menu, které slouží k nastavení prostředí, možností Builderu, klávesových zkratk, barev v editoru, ikon rychlého spouštění, fontu, priorit editoru, syntax Highlightingu, zvýraznění uživatelem nastavených slov, převedení LabWindows DOS programu pro LabWindows/CVI, generování dynamických knihoven *.dll a hlavičkových souborů.

Poslední je Help, ve kterém se nachází nápověda, vyhledávání příkladů, webové odkazy, patenty, uvítací stránka a základní informace o LabWindows/CVI.

2.4.3 Funkce NI Vision

Knihovny funkcí v LabWindows/CVI jsou zobrazeny ve stromové struktuře, jak je možno vidět v příloze na obrázku 1. Ve spodní části se zobrazují zařízení, které jsou připojeny. Pokud na nějaké dvakrát klikneme levým tlačítkem myši, můžeme ho upravovat.

Po dvojkliku levým tlačítkem myši na nějakou funkci, otevře se její funkční panel, kde jí můžeme nastavit potřebné parametry. Po kliknutí pravým tlačítkem myši na funkci je možnost zobrazení nápovědy k této funkci. Pokud klikneme pravým tlačítkem myši do oblasti s knihovnami funkcí (nebo přímo na nějakou funkci), můžeme nastavit řazení dle abecedy, nebo názvu funkce, nastavit, jestli chceme zobrazovat názvy, či popis funkcí, zobrazování funkcí hierarchicky, nebo stochasticky. Dále se zde může přidat, upravit řazení a zobrazení knihoven. Nakonec je zde ještě možnost rozbalit a sbalit všechny položky.

Standardně po instalaci obsahuje LabWindows/CVI 26 knihoven:

- User Interface Library je knihovna obsahující soubor funkcí pro ovládání GUI.
- Advanced Analysis Library nabízí komplexní sadu funkcí pro analýzu dat. Za pomoci těchto funkcí je možno jednodušeji převést data na užitečnou informaci. Knihovna obsahuje funkce které simulují a rychle a efektivně analyzují velké soubory dat.
- Formatting and I/O Library obsahuje funkce pro vstup a výstup dat a úpravu formátu dat v programu.
- V Utility Library se nachází funkce, které se nevešly do žádné z dalších LabWindows/CVI knihoven. Jsou zde různé funkce, které plní rozličné operace, jako použití systémového časovače, správa souborů a adresářů, práce s externími moduly, či pro používání více vláken.
- Knihovna ANSI C obsahuje standardní ANSI C funkce, které se mohou použít v LabWindows/CVI.
- VXI Library slouží ke kontrole VXI⁸ zařízení s vestavěným VXI řadičem, nebo počítačů vybavených MXI řadičem.
- GPIB⁹/GPIB 488.2 Library obsahuje funkce pro zařízení NI-488.2.
- RS-232 Library obsahuje funkce pro práci s zařízeními připojenými přes sériový port RS232 a National Instruments RS-485¹⁰.
- VISA Library je knihovna, poskytující VXI a GPIB vývojářům programů. Zejména se jedná o nástroje pro vývoj řadičů jednotného ovládání rozhraní vstupu a výstupu. NI Visa v současnosti podporuje VXI, GPIB, RS-232, PXI¹¹ a Ethernet.
- IVI Library obsahuje funkce programu a kontroly IVI ovladačů. Ovladače kompatibilní s IVI mají standardní rozhraní, tudíž se poté může měnit zařízení, aniž by se musel měnit Váš kód programu.

8 **VXI** je architektura sběrnice s otevřeným standardem pro automatizovaný test, založena na VMEbus.

9 **GPIB (General Purpose Interface Bus)** je univerzální měřicí systém. Jedná se také o digitální stykový systém, který umožňuje přenos dat mezi dvěma nebo více přístroji. Umožňuje připojení počítače, který může řídit tento přenos dat. Vyvinut byl v roce 1972 společností Hewlett-Packard.

10 **RS-485** se vyznačuje dvou vodičovým propojením jednotek. Maximální délka sběrnice je až 1200 m, maximální počet jednotek je 32 (dle ovladače až 128). Maximální přenosová rychlost je 200 kb/s.

11 **PXI (PCI Extensions for Instrumentation)** je modulárně nástrojová platforma zavedená v roce 1997 firmou National Instruments. PXI je podporována 54 členy aliance systému PXI (PXISA).

•TCP Support Library je knihovna funkcí, které poskytují podporu pro platformu nezávislého rozhraní pro spolehlivý orientovaný bitový proud, možností připojení do sítě TCP/IP.

•UDP Support Library obsahuje funkce, které poskytují podporu pro platformu nezávislého rozhraní pro unicast¹², broadcast¹³ a multicast¹⁴ schopného UDP.

•Internet Library obsahuje funkce ke komunikaci a přijímání souborů a příkazů ze vzdálených serverů.

•Network Variable Library obsahuje funkce pro čtení a zápis dat do síťových proměnných.

•DDE Support Library obsahuje funkce používané k propojení s ostatními aplikacemi v MS Windows pomocí DDE¹⁵ standardu.

•ActiveX Library je knihovna funkcí, která obsahuje obecné informace ActiveX automatizaci, jakož i pokyny pro používání ActiveX knihovny. Obsahuje funkce, které vytváří a kontrolují ActiveX Servery.

•DIAdem Connectivity Library poskytuje funkce, které se mohou použít k přenosu testovaných dat v National Instruments DIAdem souborovém formátu (*.TDM). Poté se mohou nahrát soubory přímo do DIAdem k analýze. Funkce této knihovny jsou určeny ke správě souborů, přidávání cest a skupin cest k souborům, ukládání dat pomocí cest, čtení datových hodnot z cest, získávání a nastavování záznamů, skupin cest a jednotlivých cest.

•TDM Streaming Library poskytuje funkce pomocí kterých lze rychle přenášet velké množství dat v souborovém formátu National Instruments TDM Streaming (*.tdms). Poté se data mohou načítat přímo do DIAdem. Tato knihovna obsahuje funkce pro správu souborů, přidávání cest a skupin cest k souboru, ukládání dat pomocí cest, čtení hodnot z cest a čtení a nastavování vlastností záznamů, skupin cest a jednotlivých cest.

•.NET library obsahuje funkce, které usnadňují volání .NET.

•Programmer's Toolbox obsahuje ovladače zařízení vybraných užitečných a vhodných funkcí. Většina funkcí ovladačů zařízení běží vícevláknově bezpečně. Nebezpečný vícevláknový běh je označen textovou zprávou na jejím panelu.

12 **Unicast** označuje v počítačové síti zasílání paketů pouze jedinému cíli (uzlu).

13 **Broadcast** je opakem unicastu, vysílá do všech uzlů v síti najednou.

14 **Multicast** rozesílá pakety pouze určité skupině stanic, něco mezi Unicast a Broadcast.

15 **DDE (Dynamic Data Exchange)** vyvinuto firmou Microsoft pro komunikaci programů mezi sebou.

•Následující čtyři knihovny patří do sady funkcí NI-IMAQ. Konkrétně se jedná o knihovny NI-IMAQ I/O, NI-IMAQ Library, NI-IMAQ for IEEE 1394, NI-IMAQdx. V této sadě se nacházejí funkce, které ovládají National Instruments plug-in IMAQ¹⁶ a RTSI¹⁷ sběrnice. V těchto knihovnách se nachází jak vysokoúrovňové I/O funkce pro maximální jednoduchost použití, tak i nízkoúrovňové I/O funkce s vysokou flexibilitou a výkonem.

•Max Configuratin Library se používá pro programování rozhraní (API¹⁸), ke kopírování konfiguračních dat do, nebo ze systému, či k vytváření zpráv o tomto systému.

2.4.3.1 NI Vision

NI Vision je poslední a nejobsáhlejší knihovnou, ve které se nachází 25 tříd. Tyto třídy obsahují funkce stejného typu.

První třídou je *Acquisition functions*, ta umožňuje vytvoření obyčejných úloh, jako je stále pořizování, sekvenční pořizování a jednorázové pořízení obrazu do NI Vision.

Funkce v *Analytic Geometry* umožňuje provádět analytické geometrické operace, jako je například získání bodů obrysu v obraze, nebo získání úhlu mezi dvěma přímkami.

Barcode I/O obsahuje funkce pro čtení různých typů čárových kódů (klasických, 2D, PDF417¹⁹, či QR²⁰).

Binary Processing obsahuje funkce ke zpracování binárních obrazů, jako například morfologické funkce, které umožňují uplatnit standardní morfologické transformace (eroze, dilatace,...).

V *Calibration* se nacházejí funkce, které umožňují prostorovou kalibraci obrazu. Prostorová kalibrace převádí pixelové souřadnice na souřadnice, které se používají ve skutečném světě.

16 **National Instruments plug-in IMAQ** jsou zařízení určené k pořizování obrazu.

17 **RTSI (Real-Time System Integration)** sběrnice používají NI-DAQ zařízení k jednodušší synchronizaci společně měřených dat.

18 **API (Aplikation Programming Interface)** je rozhraní pro programování aplikací.

19 **PDF417** neboli Portable Document File 417 je kvazidvourozměrný čárový kód, který sestává ze tří až 90 řádků, v každém z nichž může být zakódován 1 až 30 znaků

20 **QR** je dvojrozměrný kód, zapisovaného do čtverce. Ten musí mít ve třech vrcholech poziční značky ve formě soustředných čtyřúhelníků, ve čtvrtém vrcholu značku ve tvaru menšího čtyřúhelníku a ve spojnicích mezi těmito hraničními čtyřúhelníky úsečky tvořené střídavě bodem a mezerou

Caliper obsahuje funkce umožňující detekci a měření vlastností, jako jsou hrany a úhly podél cest v obraze.

V *Classification* se nacházejí funkce, umožňující identifikaci neznámých objektů, porovnáváním souborem jeho významných rysů se sadou rysů, které koncepčně představují třídy známých objektů.

Třída *Color Processing* obsahuje funkce umožňující analyzovat a zpracovávat barevné obrazy v různých barevných prostorech.

Ve třídě *Display* se nachází funkce, umožňující zobrazení obrazů v obrazových oknech.

V *Error Management* jsou funkce na hlídání chyb. Vráti chybu, s funkcí ve které se vyskytla a stanoví chybu.

V *File I/O* se nachází funkce umožňující čtení, zápis obrazu a ukládání informací na pevný disk nebo jiné paměťové zařízení.

Funkce *Frequency Domain Analysis* umožňují konvertovat obrazy mezi prostorovými a frekvenčními funkcemi a analýzu obrazu ve frekvenční oblasti.

Grayscale Processing obsahuje funkce šedotónového zpracování, pro vylepší šedotónového obrazu k dalšímu prohlížení a zpracovávání.

Funkce *Image Analysis* umožňuje vypočítat různé statistiky pixelů v obraze.

Image Management obsahuje funkce, které umožňují získání informací o obrazu, nebo manipulaci s obsahem obrazu.

Funkce v *Inspection* umožňují vytvářet šablonu a porovnávat obraz se šablonou. *LCD* funkce umožňují izolovat a číst hodnoty sedmi segmentového displeje.

Memory Management obsahuje funkci `imaqDispose`, která odstraňuje obrazy, ROI²¹, pole a zprávy a tím uvolňuje prostor který zabírají v paměti.

Funkce v *Meter* umožňuje zjištění informací o používaném měřítku a poté počítat rozměry. *OCR* obsahuje funkce optického rozpoznávání znaků a pro vývoj OCR²² čtecích aplikací.

21 **ROI (Region of Interest)**, neboli region zájmu je oblast obrazu která nás zajímá a se kterou chceme nadále pracovat.

22 **OCR (Optical Character Recognition)** je optické rozpoznávání znaků.

Funkce v *Operators* umožňují provádět aritmetické, či logické operace mezi dvěma obrazy, nebo mezi obrazem a konstantou.

Overlay obsahuje funkce, které umožňují překrývání a přidružování k obrazovému oknu.

Funkce v *Pattern Matching* umožňují vyhledávat v obraze podle šablon.

Region of Interest obsahuje funkce, které umožňují vytvářet, upravovat a extrahovat informace o oblastech zájmu (ROI).

Funkce v *Utilities* umožňují nastavení struktur, které se mohou vkládat do dalších funkcí, které odstraňují nutnost deklarování některého typu proměnných.

2.4.4 Vytvoření programu

Program lze vytvořit dvěma způsoby. Buďto se skript vytvořený v National Instruments Vision Assistant vyexportuje do LabWindows/CVI a nebo se celý program vytvoří jen v LabWindows/CVI.

Z hotového programu je možno vytvořit distribuci. Ta bude obsahovat všechny balíčky, knihovny a závislosti na aplikaci, které jsou potřebné pro běh vytvořeného programu.

3 Hodnocení kvality stírání u zkušebního okenního skla

Firma Škoda auto a.s., patřící do koncernu Volkswagen je největším výrobcem automobilů v České republice. Auta nejen vyrábí, ale i vyvíjí a testuje. V evropských automobilkách bylo po dlouhou dobu zvykem to, že si sami vyráběli většinu dílů do automobilů sami. Takovýchto vlastních dílů ale ubývá, i když jsou ještě často vyráběny přímo v areálech automobilek. Ovšem o jejich výrobu se stará externí firma. Například v Mladé Boleslavi takto externí společnosti vyrábějí pro automobily celé dveře nebo sedačky. A jak se ukazuje, v budoucnu se v automobilkách budou montovat jen velké díly, které jim vyrobí jejich externí dodavatelé. Automobilky si od toho slibují hlavně zrychlení výroby a menší závislost na dodavatelích jednotlivých součástek.

Stát se dodavatelem automobilek znamená mít odbyt svých výrobků a proto by se jím chtěla stát spousta firem. Díky tomu si automobilky mezi dodavateli mohou vybírat. Při tomto výběru rozhoduje jak cena, tak i kvalita, přičemž cena ne vždy vypovídá o kvalitě. Právě testování kvality je jedna z důležitých částí vývoje automobilu. Toto testování může ovlivnit jak kvalitu, tak i cenu výroby automobilů. Díky tomu automobilky do testování a testovacích aplikací investují nemalé peníze.

Automobil se skládá z velkého množství součástek a úspora na každé z nich může ušetřit část nákladů na jeho výrobu. Jako příklad uvedu gumičky do stěračů. Ačkoliv je tato položka ve výrobní ceně jednoho automobilu nepatrná, její špatná kvalita je celkem dobře vidět. Zanedbatelná ovšem není ve výrobě velkého množství automobilů a tak se na ní automobilky také snaží ušetřit, přičemž se snaží udržet nějaký kvalitativní standard, aby neodradili případné zákazníky.

3.1 Zadání úlohy

Ing. Jiřím Horčíčkou a Doc. Ivanem Jakschem, Csc vytvořili studii proveditelnosti kamerového systému pro objektivní vyhodnocování stírání skla automobilu. Na jejím základě bylo vytvořeno zadání aplikace, která bude hodnotit kvalitu stírání u zkušebního okenního skla.

Posouzení kvality stírání se dosud provádí manuálně ve speciálních boxech, ve kterých je provedeno cyklování stěračů. Vyhodnocení se provádí subjektivně na základě normy TL 934. Po provedení určitého cyklu stírání přijde obsluha a posoudí kvalitu stírání podle klasifikace obrazu stírání a podle tabulky. Subjektivně podle předloh ohodnotí kvalitu na stupnici od 1 do 10. To by ve firmě Škoda Auto rádi změnili. Chtěli by systém pro objektivní vyhodnocování kvality stírání, který by přesně podle zadaných norem ohodnotil kvalitu stírání (také na stupnici od 1 do 10). Požadavkem Škoda Auto je zařízení, které by mělo být mobilní, nejlépe umístitelné do současně používaných boxů. Důvodem je možnost použití zařízení na více stavech.

3.1.1 Norma

Zkouška kvality stírání se dle koncernových norem Volkswagen provádí za dobrých osvětlovacích podmínek. Provádí se s páskami stěrače a zařízením v originálním dílu karoserie (úplná přední příčná stěna karosérie, zadní víko). Nebo ve stojanu SAE. Okenní sklo se ze zadní strany podloží modrou barvou (RAL 5010). Dosedací síla se nastaví dle normy. Po záběhové době 3 až 4 cykly stírání se stírací pole kompletně ostříká vodou a kvalita stírání se posuzuje a klasifikuje po každém pohybu nahoru, eventuálně dolů.

Požadavky normy:

- Posouzení kvality stírání se provede podle klasifikace obrazu stírání podle tabulky (viz. příloha obr. 9 až 11).

- V novém stavu bříty stěrače musí dosáhnout známky nejméně 8.

- Po 200 000 cyklech stírání dosáhnout ještě známky 6.

- Po 300 000 cyklech stírání dosáhnout ještě známky 5.

- Po 500 000 cyklech stírání dosáhnout ještě známky 4.

3.1.2 Zkušební okenní sklo

Na obrázku 3.1 je vidět úplná přední příčná stěna karosérie se sklem, na kterém probíhalo testování. Na sklo se voda stříkala pomocí rozprašovače, poté se stírala a nakonec stékala do plechové vany ve které bylo celé zařízení umístěno.



Obr. 3.1: Zkušební okenní sklo

3.2 Hardwarové prostředky

Celé navrhované vyhodnocovací zařízení se bude skládat z osvětlovače, který vhodným způsobem osvětluje celou snímanou scénu, digitálních kamer s objektivy, a počítače, na kterém bude spuštěn vyhodnocovací software. Osvětlovač a kamery s objektivy budou připevněny ke konstrukci.

3.2.1 Kamery

Pro danou úlohu se použily digitální monochromatické kamery Basler scA1600-14FM (obr 3.2). Tyto kamery komunikují přes rozhraní IEEE 1394b nazývané též Firewire (datový přenos až 800 Mb/s). Přes toto rozhraní jsou i napájeny.

Nejdůležitější částí kamery je vysoce kvalitní CCD²³ senzor od firmy Sony, jehož rozlišení je 1626 x 1236 pixelů. Na základě požadované rozlišovací schopnosti 0,3 mm uvedené v normě je třeba použití 6 kamer pro vyhodnocování celé stírané plochy. Při použití 4 kamer je dosaženo rozlišovací schopnosti 0,4 mm. Ke komunikaci s kamerami

²³ **CCD** (Charge-Coupled Device) senzor je součástka pracující na principu sběru fotonů dopadajícího světla. Shromážděný náboj je poté zesílen a převeden do A/D převodníku.

přes Firewire rozhraní v systémech firmy National Instruments jsou určeny ovladače Image Acquisition Driver Software označené jako NI-IMAQdx.

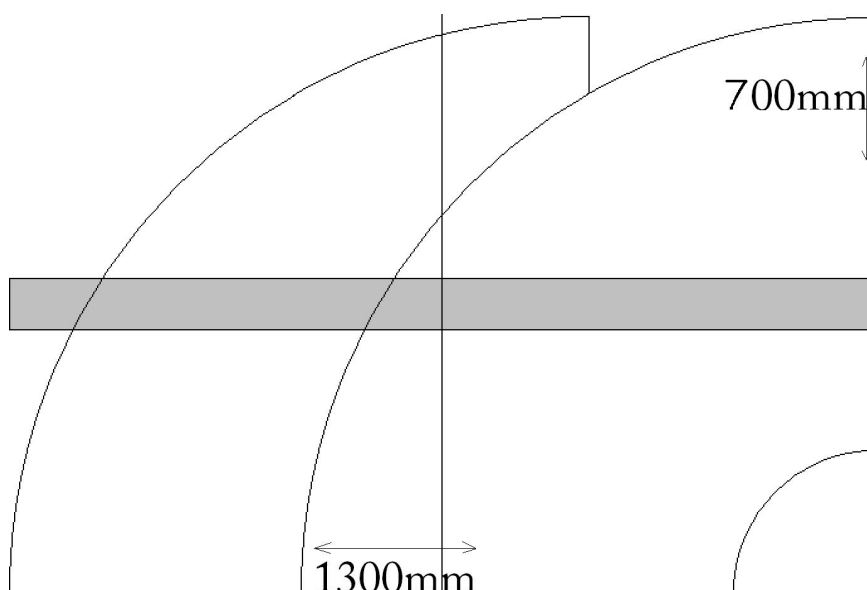


Obr. 3.2: Basler scA1600-FM

Senzor Sony ICX274 AL má rozměry 8,50 x 6,80 mm a úhlopříčku 8,923 mm (typ 1/1.8"). Každý pixel na senzoru je 4.40 μm široký a 4.40 μm vysoký. Při plném rozlišení kamera zvládne až 14 snímků za vteřinu. Expoziční doba je 100.0 μs až 24.9 ms. O převod analogového signálu na čísla se stará 12-ti bitový převodník. Vstupní napětí je 12 V a spotřeba 2,75 W. Na kameru se umisťují objektivy se závity C-Mount. Kamera o rozměrech 73.7 x 44 x 29 mm váží 160 g.

Výpočet rozlišovací schopnosti kamer je následující, podle poměru stran skla a čipu se určí strana, jejíž délka se vydělí počtem pixelů na této straně čipu (čili počet pixelů jednoho celého řádku, nebo sloupce). Poměr stran čelního skla je 13:7 a poměr čipu v kameře je 4:3.

V případě použití 4 kamer, by se rozmístily tak, jak je vidět na obrázku 3.3. Šedý pruh uprostřed označuje plochu překryvu, kterou snímají 2 kamery najednou. Při použití 4 kamer, rozmístěných tak, jak je vidět na obrázku 3.3, není jejich poměr 4:3, ale 8:6. Pokud tuto hodnotu vynásobíme 7-mi, dostaneme poměr 56:42. Poměr skla vynásobíme 6-ti a výsledkem je poměr 69:42. Díky tomu víme, že vodorovnou délku skla vydělíme počtem pixelů v 1 řádku kamery a vynásobíme počtem kamer. Čili $1300/(1626*2)$. Což je přesnost, která vyjde zhruba 0,4mm.



Obr. 3.3: Snímání 4-mi kamerami

Při použití 6-ti kamer je více možností jejich rozmístění. V tomto případě přicházejí v úvahu 2 možnosti. Buďto 3 kamery vedle sebe ve dvou řadách, nebo 2 kamery vedle sebe ve 3 řadách. Tyto možnosti se ještě zdvojnásobí v případě otočení kamery o 90°. Nejvýhodnější rozmístění kamer si tedy musíme vypočítat. Výpočet se provede vydělením délky snímání plochy počtem pixelů. Jak je vidět v tabulce 2, nejvhodnější je použít 3 kamery v řádku a 2 ve sloupci. Při tomto rozlišení kamer je přesnost 0,28mm.

Poměr stran kamery	Počet kamer v řádku	Počet kamer ve sloupci	Počet pixelů v řádku kamer	Počet pixelů ve sloupci kamer	Rozměry předního skla (mm)	Horizontální přesnost (mm)	Vertikální přesnost (mm)
4 : 3	2	3	3252	3708	1300 x 700	0,4	0,2
4 : 3	3	2	4878	2472	1300 x 700	0,26	0,28
3 : 4	2	3	1472	4878	1300 x 700	0,53	0,14
3 : 4	3	2	3708	3252	1300 x 700	0,35	0,22

Tabulka 2: Přesnost kamery

3.2.2 Objektivy

Jeden z hlavních vlivů na kvalitu obrazu, který se zobrazuje na čip kamery, má objektiv (obr 3.4). Objektiv je složen z několika čoček do skupin, které mohou být z různých materiálů (skel), ale taktéž mohou mít různé tvary, velikosti a indexy lomu. Jsou opticky centrované, což znamená, že jsou sestaveny v optické ose objektivu. Pohybují se během ostření, či zoomování (změna ohniskové vzdálenosti). Při většině aplikací počítačového vidění se používají objektivy o vysoké kvalitě, které nemají zoom. Tak tomu je i při této aplikaci.



Obr. 3.4: Objektiv

Při volbě objektivů je hlavním kritériem snímací vzdálenost a velikost snímané scény, které určují vzdálenost kamer od čelního skla. Jednotlivé typy objektivů jsou uvedeny v tabulce 3. Vzhledem k prostorové omezenosti boxů je potřeba použít objektivy s ohniskovou vzdáleností kratší než osm milimetrů. Jejich nevýhodou je značné zkreslení. Pro odstranění případných odlesků je vhodné použití polarizačních filtrů²⁴.

ohnisková vzdálenost f [mm]	snímací vzdálenost [mm]
4	271
6	406
8	541
12	812
16	1082

Tabulka 3: Objektivy

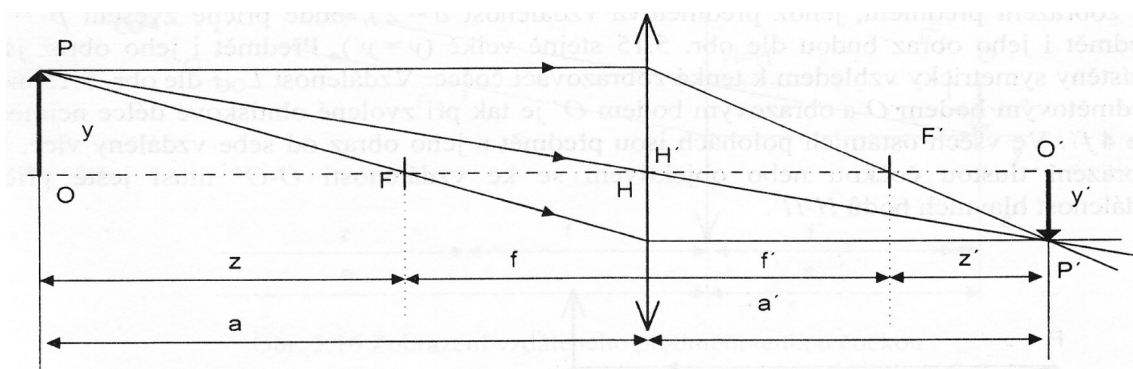
²⁴ **Polarizační filtr** propouští elektromagnetické záření (světlo), které kmitá v jednom směru. Požadovaný směr kmitání světla je ovlivněn natočením filtru.

Pro výpočet snímací vzdálenosti byl použita upravená zobrazovací rovnice:

$$a = \frac{f}{y/y'}$$

Vzorec 1: Zobrazovací rovnice

kde a je snímací vzdálenost, f je ohnisková vzdálenost, y' je velikost obrazu a y je velikost předmětu. K odvození vzorce byla použita zobrazovací rovina (obr 3.5).



Obr. 3.5: Zobrazovací rovina

3.2.3 Osvětlení

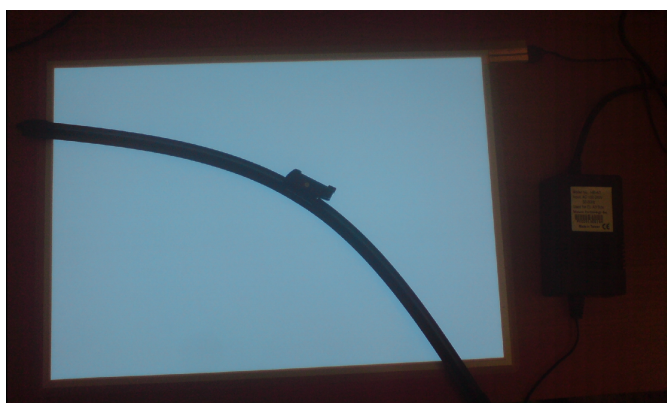
Jedním z nejdůležitějších zařízení pro spolehlivé vyhodnocení kvality stírání je osvětlovač. Ten musí zaručit rovnoměrné difúzní světlo po celé ploše stírané plochy čelního skla automobilu. Musí také zajistit co největší změnu intenzity světla pro hledané poruchy stírání. Pokud by v cestě mezi kamerou a sklem byly i jiné okolní zdroje světla, stíny, nebo nebylo zajištěno dostatečné rovnoměrné osvětlení, bylo by velmi obtížné tyto nedostatky opravit a ovlivnilo by to vyhodnocování.

Prvním typem osvětlení byla soustava zářivek, jejichž světlo bylo rozptýleno použitím velkého difuzoru pro celé přední sklo. Okolní osvětlení je potřeba v nejvyšší možné míře potlačit umístěním čelního skla s osvětlovačem ve vhodném prostředí. I tak bylo na základě laboratorních experimentů toto osvětlení vyhodnoceno jako nevyhovující.

Dalším typem, který byl nakonec zvolen, bylo osvětlení pomocí elektro luminiscenčních panelů (EL panelů). Jedná se o velmi tenké (slabší než 0,6mm), ohebné svítící fólie. Vedle malé tloušťky a váhy poskytují jasné, měkké světlo s širokoúhlým zobrazením.



Obr. 3.6: Vypnuté ELP-A4 a zdroj

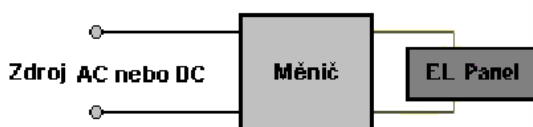


Obr. 3.7: Zapnuté ELP-A4, stěrač s gumičkou a zdroj

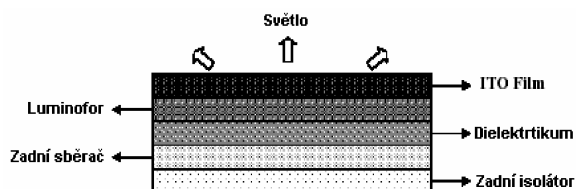
Klasické žárovky a diody dodávají bodové osvětlení, fluorescenční a neonové lampy dodávají čárové světlo, ale EL panely generují rovnoměrné, měkké, plynulé, plošné osvětlení po celé osvětlovací ploše. Neprodukují žádné záření, jelikož světlo vzniká elektro-chemicky. Barva osvětlení je bílá a jas může být v rozmezí 200 až 1000 Luxů. Panely mají navíc malou spotřebu a neohřívají se (95% energie se přímo přemění na světlo).

Při této úloze byly použity panely ELP-A4. Jak již název napovídá, jedná se o formát A4, tedy 307 x 220 mm. Tento formát je i včetně vnějšího rámečku, ve kterém

je umístěn vodič, který obepíná celou osvětlovací plochu. Jak je vidět i na obr. 3.6 a 3.7, tento rámeček nesvítí. V rohu panelu je umístěn konektor pro připojení vodiče vedoucího z měniče. Schéma napájení EL panelů je vidět na obr. 3.8. EL panely pracují pod určitým napětím a frekvencí, proto se napájecí napětí, stejnosměrné nebo střídavé (230 VAC, 110 VAC, 12/24 VDC), konvertuje měničem. V úloze je použito napájení z elektrické sítě, čili 230 VAC. Proudová zatížitelnost je v rozsahu 0,1 až 0,4 A/cm² (jas závisí na spotřebě proudu), spotřeba energie je v rozsahu 0,01 až 0,08 W/cm² (opět podle jasu). Proudové zatížení jednoho ELP-A4 je tedy v rozsahu 0,6 až 2,7 A a spotřeba energie je v rozsahu 0,06 až 0,54 W. Při použití sedmi panelů o jasu 1000 Luxů je tedy proudové zatížení 18,9 A a spotřeba energie 3,78 W. Životnost EL panelů závisí na teplotě a vlhkosti prostředí. Za běžných podmínek je kolem 2 až 5 tisíc hodin, při použití s vysokým jasnem, ale při použití s nižším jasnem může vydržet 5000 až 10000 hodin, i déle.



Obr. 3.8: Napájení EL panelů

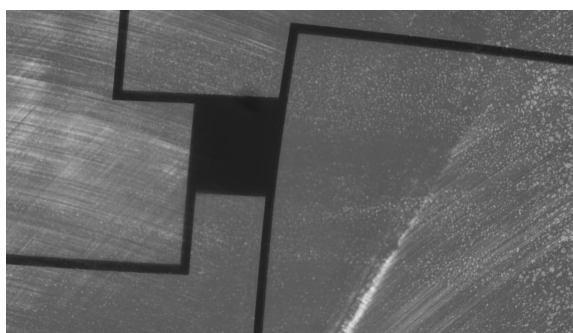


Obr. 3.9: Struktura EL panelů

EL panel má 5 různých vrstev, jak je vidět na obr. 3.9. První vrstvou je zadní izolátor, druhou vrstvou je zadní sběrač, třetí vrstvou je dielektrikum, což je izolant schopný polarizace. Čtvrtou vrstvou je luminofor, což je látka která je schopná uchovávat dodávanou energii a poté ji přeměnit na světelnou energii a pátou vrstvou je ITO²⁵ film, sloučenina india, cínu a kyslíku a jedná se o velmi tenkou bezbarvou průhlednou fólii. ITO slouží jako anoda. Izolační vrstvy v EL panelech jsou nezbytné pro zabránění elektrickému oblouku a následnému jiskření mezi dvěma vodivými vrstvami.

²⁵ ITO (Indium, Tin, Oxide) se používá hlavně jako průhledný vodivý povlak, např. na displeje z tekutých krystalů, plazmové displeje, dotykové panely, solární články...

Díky velmi nízké hmotnosti panelů bylo možné snadné umístění do konstrukce. Panel se přilepil za pomoci oboustranné lepicí pásky. Bohužel panely mají okolo sebe vodič který nevytváří světlo (viz. Obr. 3.10) a velikost A4 byla maximální velikostí, která se dala zakoupit, proto i po seskládání EL panelů vedle sebe jsou na skle viditelné tmavé obdélníky neosvětlených míst. Do EL panelů je možné vytvořit kruhové otvory pro kamery, které se budou na sklo dívat. Tyto otvory taktéž nebudou vytvářet světlo a tak se na skle objeví další neosvětlené plochy, tentokrát kruhové. Do panelů lze vytvořit i více otvorů, ale tento krok už nelze vzít zpět, jelikož tato místa už nebudou nikdy svítit, proto probíhalo testování tak, že se okolo kamery (na desku) přilepily na sebe poskládané desky (jak je vidět na obr. 3.10). Skrz černý čtverec snímala kamera sklo a černé pruhy jsou vnějšími rámečky panelů, ve kterých se nachází vodič. Do něj se na rozdíl od osvětlovací části panelu nesmí dělat otvory. A ačkoliv se jedná o plošné stejnoměrné osvětlení na skle jsou vidět světlejší a tmavší místa, za které mohlo jak špatné nanesení podkladové barvy, tak i značné poškrábání skla.



Obr. 3.10: Osvětlení pomocí EL panelů

3.2.4 Konstrukce

Jedná se o speciální mechanickou konstrukci uchycení osvětlovačů a kamer s objektivy. Konstrukce je přenositelná a lze ji umístit na více typů předních částí karosérií.

3.3 Aplikace Hodnocení kvality stírání u zkušebního okenního skla

Aplikace byla vyvíjena v prostředí LabWindows/CVI. Nápomocné bylo také NI Vision Assistance, ve kterém se obraz předzpracovával a zkoušely se různé funkce ke zpracování a předzpracování.

Aby bylo co zpracovávat, musí se nejdříve obraz načíst z kamery. K tomu se používá knihovna NIMAQ. Pro načtení obrazu byla použita funkce `imaqEasyAcquire`.

Image imaqEasyAcquire(const char* interfaceName);*

Ta nastaví zařízení určené pomocí *InterfaceName* (např. *img0*, či *img1*) k pořízení obrazu, poté jeden obraz načte a vrací získaný obraz. V případě neúspěchu funkce vrací NULL.

3.3.1 Testované metody

Testované metody se dají rozdělit do dvou skupin, a to na metody s použitím segmentace a metody bez použití segmentace.

Díky celoplošnému stejnoměrnému osvětlení a díky tomu že voda která zbude na skle po setření má o něco menší hodnotu jasu (je tmavší) nežli okolní plocha(pozadí) je možné použít segmentaci prahováním. Ta nám má vysegmentovat objekty (vodu) od pozadí (setřeného skla). Prahování se používá především jako metoda předzpracování a tak je tomu i zde. Jedná se velmi rychlou metodu.

V případě ideálních obrazů by se dal po prahování obraz rozdělit pomocí ROI. Poté by se chyba dalo zařadit kam chyba patří, zobrazili by se regiony s chybou a poměry chyb k celkové ploše (v procentech). Bohužel zpracovávaný obraz nebyl ideální.

Proto byla zvolena další metoda zpracování. Vzhledem k tvaru vad (kruhové výseče) byla zvolena Houghova transformace. I v případě, že se vadu nepodaří vysegmentovat do jednoho objektu, transformace to vyhodnotí jako jednu vadu. Tato

metoda je do značné míry odolná proti šumu a jiným menším vadám. Problémem je použití této transformace pro celý obraz a to hlavně ze dvou důvodů. Prvním je velká náročnost výpočtu a druhým rozdíly ve velikostech hledaných objektů. Malé objekty se na rozdíl od těch velkých nepodaří v obrazu nalézt a to vzhledem k menšímu lokálnímu maximu v akumulátoru.

Další metodou je průměrování, které se používá bez předchozí segmentace. Tato metoda potřebuje rovnoměrné osvětlení. V případě že by osvětlení nekopírovalo tvar autoskla, v okrajích by nebylo tak intenzivní. Stíraná plocha se rozdělí na kruhové výseče (ROI), které se budou překrývat. V každém regionu vypočítá průměrnou jasovou hodnotu a pokud bude průměrná hodnota nižší, než je průměrná hodnota pozadí, bude tato oblast označena za vadnou. Jedná se o poměrně jednoduchou a rychlou metodu k nalezení vad, ale nevyhodnocuje jejich velikost. Metoda by se dala zpřesnit Houghovou transformací.

A právě kombinace těchto metod se zdála být nejlepším řešením. Na obraz by se nejdříve použila průměrovací metoda, která by našla oblasti s chybami. Na tyto oblasti by se poté použilo prahování a následně Houghova transformace, která by jednotlivé chyby označila. Podle množství a velikosti chyb by se pak objektivně ohodnotila kvalita stírání.

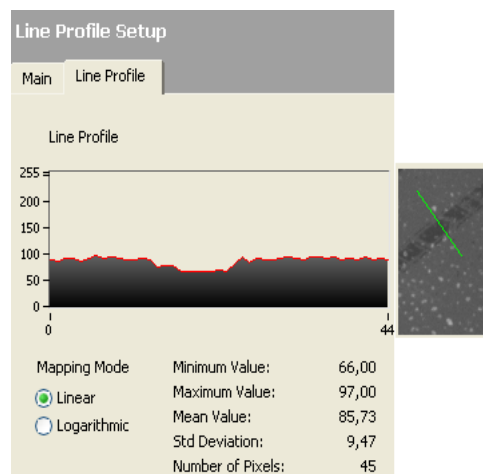
Bohužel ani tato kombinace metod nefungovala podle představ. Viníkem byly hlavně tyto tři aspekty:

- 1.Špatné nanesení podkladové barvy skla automobilu (nekonstantní jas pozadí – na obrazu byli bubliny a pruhy).

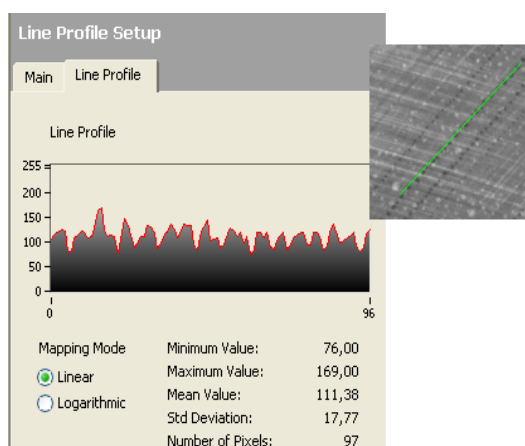
- 2.Velké poškrábání skla automobilu (jednalo se o kruhové výseče v místech kde se měla hodnotit kvalita stírání).

- 3.Napájecí vodič kolem EL panelů a otvory pro kamery tvořil tmavá místa (tato místa nebyla osvětlena) - nejmenší vliv, který se dá vyřešit tím, že by se tato místa neuvažovala.

Na obrázku 3.11 je vidět ideální stav, kdy na světlejším pozadí je tmavší voda. Pomocí *Line Profile* jsou zobrazeny hodnoty šedi pixelů pod zelenou čarou. Průměrná hodnota okolí je zde kolem 90-ti a průměrná hodnota jasu vody je kolem 70-ti.



Obr. 3.11: Line Profil 1



Obr. 3.12: Line Profil 2

Problém nastává například na poškrábaném skle, jelikož poškrábaná místa byla světlejší než ta nepoškrábaná. Průměrná hodnota na těchto místech s vodou je skoro stejná, jako na nepoškrábaném místě bez vody, jak je vidět na obrázku 3.12.

I díky tomu se přišlo na následující metodu, diferenci. Nejprve se načte obraz čistého skla před setřením a poté po testovacím setření. Tyto dva obrazy se od sebe odečtou a rozdíl (difference) mezi nimi je ona chyba (voda která po setření na skle zůstala). Tato metoda je pro tento případ ideální, jelikož zanedbává vady skla, podkladové barvy. Navíc je velmi rychlá a výpočetně nenáročná. Jediným problémem

by mohlo být posunutí kamery např. vibracemi. Při testování k tomu nedošlo, ale pokud by k tomu docházelo, dalo by se to napravit např. za pomoci referenčních značek.

3.3.1.1 Prahování

Jedná se o nejjednodušší segmentační postup. Prahování vychází z toho, že mnoho objektů, nebo oblastí obrazu má konstantní odrazivost, či pohltivost povrchu. Poté se může využít určené jasové konstanty (prahu), k oddělení objektů od pozadí. Jedná se o nejstarší segmentační metodu, která je díky své jednoduchosti i nejrychlejší a lze ji provádět v reálném čase. Často se také využívá k předzpracování, např. při detekci hran v obraze, či rozpoznávání znaků.

Prahováním se tedy nazývá transformace vstupního obrazu f na výstupní binární obraz g daná vztahem

$$g(i, j) = 1 \text{ pro } f(i, j) \geq T, \quad g(i, j) = 0 \text{ pro } f(i, j) < T,$$

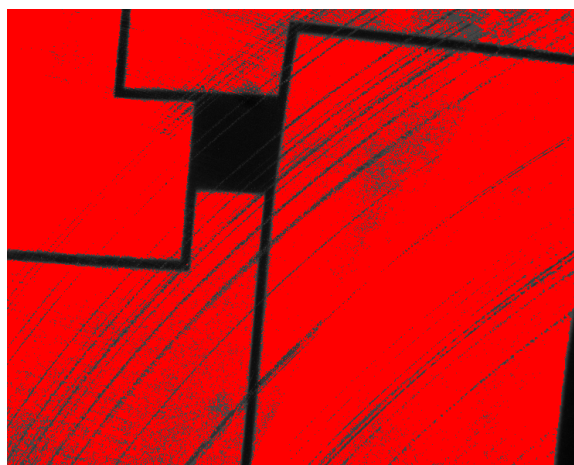
Vzorec 2: prahování

kde T je předem určená konstanta (práh). $g(i, j) = 1$ pro obrazové elementy po segmentaci náležící objektům a $g(i, j) = 0$ pro obrazové elementy po segmentaci náležící pozadí (nebo opak). Klíčovým úkolem při segmentaci prahováním je určování vhodného prahu. Většinou není možné použít stejný práh pro celý obraz. Dokonce ani v jednoduchých případech nemusíme za pomoci globálního prahování s jediným prahem dosáhnout požadovaných výsledků. Může to být způsobeno například různými vlastnostmi snímacího zařízení v celé ploše obrazu, nebo nevhodným osvětlením které způsobuje změnu jasů objektů i pozadí. Při těchto situacích může pomoci prahování s proměnným prahem, u kterého je práh určován podle lokálních vlastností obrazu.

Procentní prahování využívá apriorní znalosti poměru ploch objektů a pozadí. Víme-li, že objekty zaujímají $1/p$ plochy obrazu, na základě histogramu snadno určíme takovou hodnotu prahu T , aby právě $1/p$ plochy měla úroveň jasu menší než T . Složitější metody se opírají o analýzu tvaru histogramu.

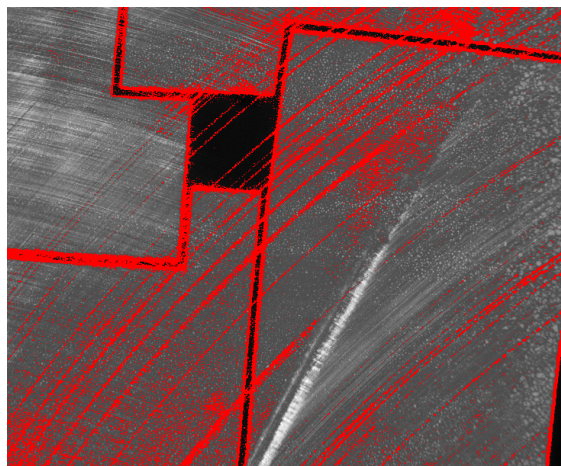
Prahování bylo testováno v NI Vision Assistant. Zde jsou 3 typy prahů: manuální, lokální a automatický. Dále je zde na výběr vyhledávání světlých či tmavých objektů (u manuálního prahování i šedých objektů). Segmentace pixelů se provádí na šedotónových obrázcích.

Manuální prahování umožňuje vybrat rozsah hodnot jasu pixelů. Lokální prahovací operace zvolí pixely pomocí místně adaptivních prahovacích algoritmů. Lokální prahování se používá pro aplikace, v jejichž obrazech se projevují nejednotné světelné změny, jejichž výsledkem je nestejně osvětlení, či stíny. Automatické prahové operace zvolí hodnotu prahu samy. Automatický práh se používá, pokud je očekáváno jednotné osvětlení obrazu.

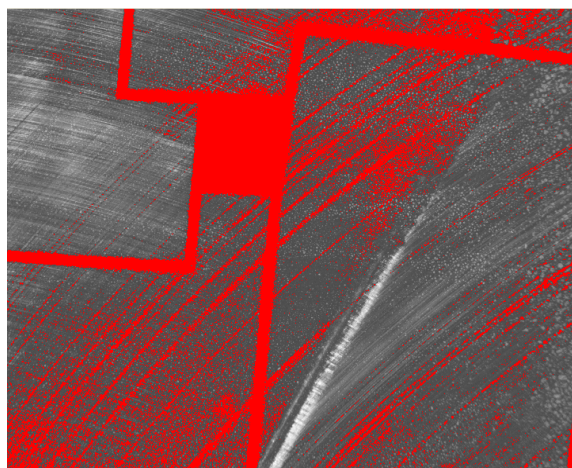


Obr. 3.13: Manuální prahování, světlé objekty

Při hledání světlých objektů (obr. 3.13) oddělíme pixely, které mají jasovou hodnotu od uživatelem definované hodnoty do 255-ti. Při hledání tmavých objektů (obr. 3.15) oddělíme pixely, které mají jasovou hodnotu od 0 do uživatelem definované hodnoty. Při hledání šedých objektů (obr. 3.14) oddělíme pixely, které mají jasovou hodnotu v rozmezí mezi dvěma uživatelem definovanými hodnotami.



Obr. 3.14: Manuální prahování, šedé objekty



Obr. 3.15: Manuální prahování, tmavé objekty

Po prahování vznikne binární obraz. Všechny pixely, které nespádají do příslušného rozsahu se přepíše na 0 a pixely které do tohoto rozsahu spadají se přepíše na 1.

NI Vision Assistant obsahuje dvě metody lokálního prahování, Niblack (viz. přílohy obr. 2) a Background (viz. přílohy obr. 3). U těchto metod se používá jádro, které nám udává velikost okolí každého pixelu, používané pro výpočet prahu. Velikost okolí bývá obvykle stejně velká jako je velikost hledaného objektu, který chceme prahem izolovat. Niblack počítá hodnotu prahu pro každý jednotlivý pixel. Je založena na statistice okolních pixelů. Tato metoda pomáhá kompenzovat velké osvětlení v obraze. Další možností u této metody je Deviation Factor, který určuje citlivost

algoritmu. Nastavuje se zde hodnota od 0 do 1, přičemž 0 je nejvíce citlivá na šum. Čím nižší je hodnota odchylky, tím menší hodnotu musí mít průměrná hodnota vybrané oblasti. Druhou metodou je Background Correction, která vykonává korekce pozadí k odstranění nejednotných světelných efektů a potom provádí prahování pomocí mezitřídnicích variací prahovacího algoritmu.

Automatických metod prahování je v NI Vision Assistant pět. První z nich, Clustering (viz. přílohy obr. 4) používá prahování obrazu založené na statistické technice nazývané seskupování. Druhou metodou je Entropy (viz. přílohy obr. 5), která používá prahování obrazu založené na jeho analýze, technikou zvanou entropie. Metric (viz. přílohy obr. 6) je následující metoda, používající k prahování obrazu výpočet optimálního prahu, který závisí na povrchu představujícího původní šedotónovou hodnotu pomocí metrické techniky. Moments (viz. přílohy obr. 7) je další metodou, která používá prahování obrazu za pomoci statistických nástrojů nazývaných momentové, které přepočítávají teoretický binární obraz. Poslední automatickou prahovací metodou v NI Vision Assistant je Inter Variance (viz. přílohy obr. 8). Prahování obrazu je zde založeno na klasické statistické metodě zvané směrodatná odchylka.

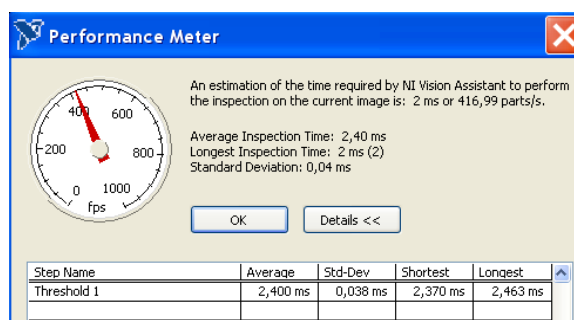
Ve vytváření programu byla použita funkce `imaqThreshold` z knihovny Vision:

```
imaqThreshold(imageArr[vystup], imageArr[vstup], mim_Prah, max_Prah, TRUE, 1);
```

V případě neúspěchu je návratovou hodnotou funkce 0, jinak funkce vrací nenulovou hodnotu. Prvním parametrem je výstupní obraz a druhým parametrem vstupní obraz. Dalšími parametry jsou, spodní část prahu, vrchní část prahu, poté parametr *TRUE*, či *FALSE*, podle toho, jestli chceme hodnoty které se nacházejí v rozmezí prahu přepsat či nikoliv. Posledním parametrem je hodnota, ne kterou se případně přepíše hodnoty, které jsou v rozmezí prahu.

Průměrná doba prahování testovaných obrazů o rozměrech 1600 x 1200 pixelů je 2,400 ms, jak je vidět na obr. 3.16. Údaj byl naměřen pomocí *Performace Meter*,

který je součástí NI Vision Assistant. Nejrychlejší čas je 2.370 ms a nejpomalejší čas je 2,463 ms. Standardní odchylka je tedy 0.038 ms.



Obr. 3.16: Doba vykonávání prahování

3.3.1.2 Houghova transformace

Jedná se o metodu sloužící k hledání definovaných objektů v obraze. Klasická Houghova transformace slouží především k nalezení pravidelných objektů (úsečky, elipsy, kružnice, atp.), jelikož vyžaduje parametrický popis objektu. Pokud není možný jednoduchý analytický popis objektu, používá se zobecněná Houghova transformace.

Houghova transformace se se využívá například v lékařství. V těchto aplikacích je transformace realizována především k hledání hranic objektů v podobě křivek a úseček. Největší výhodou Houghovy transformace je určitá tolerance rozdílů rozpoznávaných objektů, oproti jejich parametrickému popisu, a relativní odolnost proti zašumění vstupního obrazu.

Nejjednodušším případem je hledání úseček, resp. přímek. Klasickou rovnicí přímky

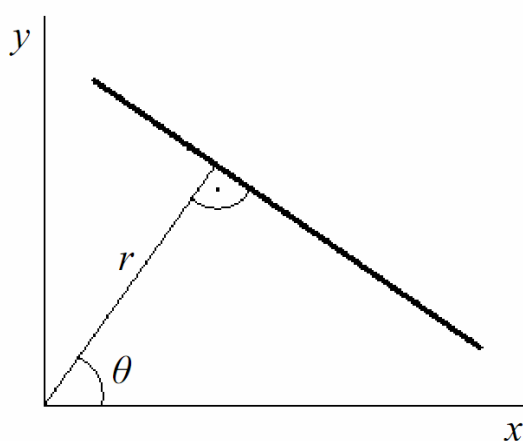
$$y = k * x + q$$

Vzorec 3: rovnice přímky

nelze použít. Protože parametr k definující sklon, tedy směrnici přímky je neomezený. Jako parametrický popis se proto použije polární tvar rovnice přímky

$$x \cdot \cos\theta + y \cdot \sin\theta = r.$$

Vzorec 4: polární tvar rovnice přímky



Obr. 3.17: Parametricky popsaná úsečka

Jak je vidět na obrázku 3.17, parametr r představuje vzdálenost přímky od počátku a úhel θ svírají osa x a kolmice vztyčená k přímce procházející počátkem. Intervaly těchto parametrů jsou omezené pro celou množinu všech přímek. Jsou dva způsoby volby těchto dvou intervalů. První možností je interval $<0;360)$ pro úhel θ , kdy parametr r nabývá pouze kladných hodnot (teoreticky až do velikosti úhlopříčky obrázku, kterou lze spočítat pomocí Pythagorovy věty). Druhou možností je uvažování i záporných hodnot parametru r , čímž je interval úhlu θ zredukován analogicky na polovinu, tedy $<0;180)$.

U této metody se nejdříve nadefinuje prostor obsahující všechny možné hodnoty parametrů hledaného objektu. Tomuto prostoru se říká akumulátor. Počet parametrů objektu nám určuje dimenzi matic akumulátoru. Hodnoty v akumulátoru musí být na

počátku shodné (většinou 0). Například úsečka má parametry r a θ , které představují souřadnice akumulátoru.

Poté se projde celý obraz, pixel po pixelu a pokud tento pixel náleží objektu dosadí se do vzorce 4. Za x a y se dosadí souřadnice nalezeného bodu. Všechny hodnoty intervalu vycházející z akumulátoru se dosazují za jednu z neznámých r , θ a druhá hodnota se dopočítá. Pro dvojici získaných hodnot $[r, \theta]$ se přičte konstanta (obvykle 1) na příslušné místo do akumulátoru. Jelikož se hodnoty takto akumulují je matice parametrů nazývána akumulátor.

Nakonec se vyberou lokální maxima, definující hledané objekty. Zjednodušeně se dá říct, pokud se v obraze nachází více objektů s různými velikostmi, jsou nalezeny ty objekty, které tvoří více pixelů, jelikož jim odpovídá větší maximum v akumulátoru.

K nalezení kružnic se použije analytický popis kružnice

$$(x - a)^2 + (y - b)^2 = r^2$$

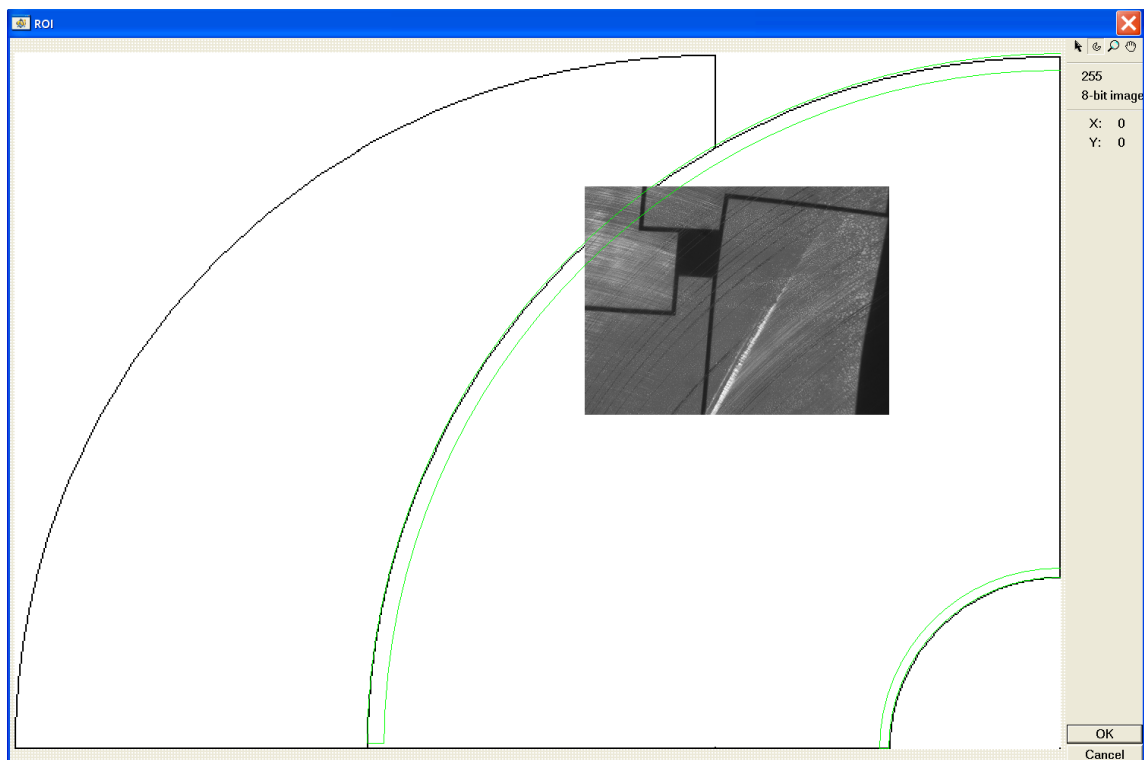
Vzorec 5: Analytický popis kružnice

Počet neznámých je nyní tři, proto je akumulátor třírozměrný. Hodnoty $[a, b]$ jsou souřadnicemi středu kružnice a r je poloměr. Intervaly hodnot parametrů se poté volí dle požadavků. Dají se nalézt i kružnice, které nejsou v obraze celé, což je pro hledání zbytku tekutiny na skle vhodné. Princip je stejný jako u hledání přímek. Projde se obraz, hodnoty se dosadí do rovnice a poté se přičte konstanta na příslušnou pozici do akumulátoru. Při zvýšení parametru o jeden se výpočetní náročnost zvýšila o řád.

3.3.1.3 Průměrná hodnota jasu v oblasti

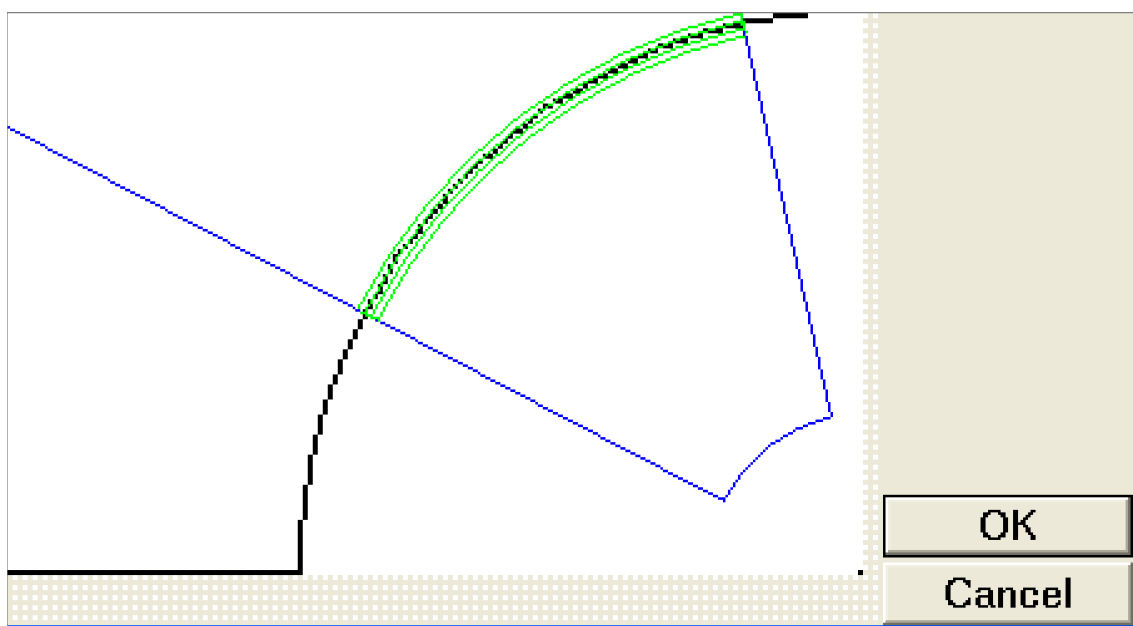
Pokud je gumička stěrače automobilu poškozena, zanechává na skle stopy kruhové výseče. Ta má střed v místě uchycení stěrače k automobilu. Voda je na obraze tmavší než okolí. Pokud by jsme si uložili průměrnou hodnotu okolí a rozdělili si plochu kterou budeme testovat na regiony zájmů (ROI), jak je vidět na obrázku 3.18, měli by oblasti

obsahující tmavší průměrnou hodnotu jasu, než oblasti ve kterých by se voda nenacházela.



Obr. 3.18: Okno programu ve kterém se nastavovali ROI

K testovacím účelům byl napsán program v LabWindows/CVI. V tomto programu se po kliknutí na tlačítko *prumerovani* otevřelo okno (obr. 3.18), kde se pomocí myši zadal počáteční a koncový region zájmů (oblast ohraničený zelenou čarou). Program poté rozdělil oblast mezi těmito oblastmi na stejně široké oblasti, které se překrývaly. Nejdříve se vypočítala průměrná jasová hodnota celé vybrané oblasti a poté se počítaly průměrné jasové hodnoty pro jednotlivé oblasti na které byla celá oblast rozdělena. Pokud se průměrná jasová hodnota nějaké takové oblasti lišila o odchylku více, než je průměrná jasová hodnota celé oblasti, byla tato malá oblast zobrazena v obraze (obr. 3.19).



Obr. 3.19: Část okna s nalezenými oblastmi

Pro práci s oblastmi byla použita funkce *imaqROI*. U každé této oblasti se vytvořila maska právě za pomoci ROI. Následně se do struktury typu *quantify* uloží vypočítané statistické parametry obrázku. Poté se z této struktury načte střední hodnota, pomocí funkce *imaqQuantify*, jejímiž vstupními parametry jsou obraz a maska.

```
mask = imaqCreateImage (IMAQ_IMAGE_U8, 2);
```

```
imaqROIToMask(mask,region, 1, NULL, NULL);
```

```
quantifyRpt_Ptr = imaqQuantify(image, mask);
```

Pokud je střední jasová hodnota daného regionu větší než-li *max*, nebo menší než-li *min*, je tato oblast zakreslena do obrazu pomocí funkce *imaqOverlayROI*. Dále se vytvoří

bod funkcí *imaqMakePoint*, který slouží následující funkci *imaqOverlayText* k zapsání textu k příslušné oblasti.

```
if(quantifyRpt_Ptr->global.mean < min || quantifyRpt_Ptr->global.mean > max){  
  
    imaqOverlayROI (image, region, IMAQ_POINT_AS_CROSS, NULL, NULL);  
  
    point = imaqMakePoint(x, y);  
  
    imaqOverlayText(image, point, "Popis", &barva, &overlayTextOptions, NULL);  
  
}
```

3.3.1.4 Diferenční metoda

Tato metoda patří mezi operátory, které provádí aritmetické a logické operace s obrazem. Diferenční metoda (obr. 3.20) může odečítat konstantu nebo obrázek od původního vstupního obrazu. Jedná se o takzvanou operaci pixel-na-pixel, kdy se od každé hodnoty pixelu jednoho obrazu odečte hodnota pixelu z druhého obrazu se stejnými souřadnicemi.

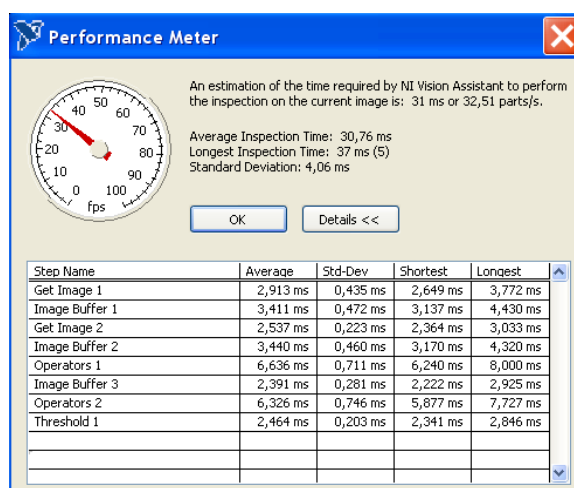


Obr. 3.20: *Diference, od obrazu s chybou se odečte obraz bez chyby a výsledkem je hledaná chyba (zvýrazněná pomocí prahování).*

Pro danou metodu je v knihovně Vision funkce `imaqSubtract`, která v případě chyby vrací nulu. Funkce má 3 vstupní parametry, konkrétně se jedná o 3 ukazatele na obraz. Prvním je výstupní obraz, a další dva jsou obrazy mezi kterými se provede difference.

`imaqSubtract(imageArray[vystup], imageArray[vstup1], imageArray[vstup2]);`

Jak je vidět na obrázku 3.21, diferenční metoda se neskládá z jedné funkce, ale z více funkcí, jejichž standardní celkový čas je 30,76ms. Nejdřív se načte první obraz, který se uloží, poté se načte druhý obraz, který se taktéž uloží. Poté se provede difference a nakonec se provede prahování. Rozdíly difference nejsou totiž natolik veliké, aby byly zřetelné na obraze. Za pomoci prahování se tyto části zvýrazní. Časy jednotlivých funkcí jsou vidět na obr. 3.21.



Obr. 3.21:Doba vykonávání prahování

3.3.2 Program

Program byl vytvářen ve výše zmiňovaném LabWindows/CVI. Nejedná se o finální verzi programu určenou k předání firmě Škoda Auto a.s., ale spíše o testovací verzi programu ve které se testovaly především dříve představené funkce. V tomto programu je zatím pouze testována funkčnost a použitelnost jednotlivých metod a jejich vzájemné propojování.

Níže budou představeny důležité části zdrojového kódu spolu s grafickým uživatelským rozhraním. Dále zde bude stručný popis programu společně s jednoduchým návodem.

3.3.2.1 Zdrojový kód

Program se skládá ze 4 souborů, jímž jsou 2 hlavičkové soubory *hks.h* a *EasyAcquire.h*, *hks.c* soubor psaný v jazyce C a *hks.uir* který obsahuje data grafického uživatelského rozhraní.

Soubor *hks.h* slouží především k definici ovládacích prvků prvků grafického uživatelského rozhraní a inicializaci funkcí které jsou v *hks.c*. K definování komunikace s připojenými kamerami se používá *EasyAcquire.h*, která byla vložena z NI Vision. Hlavičkové soubory v tomto programu tedy plní podporu komunikace programu s okolními soubory, či kamerami.

Soubor *hks.uir* byl vytvořen především grafickým editorem v prostředí LabWindows/CVI. Samotné grafické uživatelské rozhraní je popsáno v následující kapitole.

Nejdůležitější část programu byla napsána v jazyce C a je uložena v souboru *hks.c*. Na jeho počátku se načítají knihovny, hlavičkové soubory a inicializují globální proměnné. V další části souboru jsou funkce. První funkcí je *main*, která je spuštěna po startu programu. Ta zobrazí panel určený k ovládání programu uživatelem pomocí funkce *DisplayPanel* z knihovny *NI Vision*. Jejím parametrem je *panelHandle*. Tato funkce spustí pomocí *RunUserInterface* uživatelské rozhraní ve kterém se spouštějí uživatelem zvolené funkce. Pokud se spustí *QuitUserInterface*, které může být umístěno ve funkci která se spouští tlačítkem quit, jak je tomu i v tomto programu, ukončí se *RunUserInterface*, případně i jeho zpracovávání, ale neukončí samotný program. Ten v tomto programu skončí až po skončení funkce *main*, před jejímž koncem se nachází *DiscardPanel*. Tato funkce slouží k ukončení ovládacího panelu a jeho smazání z paměti. Jejím parametrem je *panelHandle*.

Dále program obsahuje funkce pro načítání obrazu (obr 3.22). Obraz se načítá jak z kamer, tak i ze souboru. Přičemž se později bude načítat obraz jen z kamer a načítání obrazu ze souboru slouží jen k testování. Jak již bylo dříve zmíněno, obraz se načítá z kamer za pomoci *imaqEasyAcquire*, jehož parametrem je název kamery. Data z kamer (obraz) se načtou do obrazových polí, z každé kamery do jednoho příslušného pole. Na konci se ještě provede rozsvícení diod na panelu, které označuje správné načtení, nebo případnou chybu při načítání.

```
//----- Nacte vzor z kamer -----//
int CVICALLBACK NactiKamery2Callback (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            imageArray[NUM_IMAGES + 0] = imaqEasyAcquire ("img0");
            imageArray[NUM_IMAGES + 1] = imaqEasyAcquire ("img1");
            imageArray[NUM_IMAGES + 2] = imaqEasyAcquire ("img2");
            imageArray[NUM_IMAGES + 3] = imaqEasyAcquire ("img3");
            imageArray[NUM_IMAGES + 4] = imaqEasyAcquire ("img4");
            imageArray[NUM_IMAGES + 5] = imaqEasyAcquire ("img5");
            /*
            imaqDisplayImage (imageArray[NUM_IMAGES + 0], 0, 1);
            imaqDisplayImage (imageArray[NUM_IMAGES + 1], 1, 1);
            imaqDisplayImage (imageArray[NUM_IMAGES + 2], 2, 1);
            imaqDisplayImage (imageArray[NUM_IMAGES + 3], 3, 1);
            imaqDisplayImage (imageArray[NUM_IMAGES + 4], 4, 1);
            imaqDisplayImage (imageArray[NUM_IMAGES + 5], 5, 1);
            */
            SetCtrlAttribute (panelHandle, PANEL_LED2, ATTR_ON_COLOR, VAL_RED);
            SetCtrlVal (panelHandle, PANEL_LED2, VAL_BLUE);
            break;
    }
    return 0;
}
```

Obr. 3.22: Funkce která načítá obraz z kamer do obrazových polí

Program obsahuje dvě takovéto funkce *NactiKameryCallback* a *NactiKamery2Callback*, které jsou spuštěny událostí která vznikne po kliknutí na příslušná tlačítka v uživatelském okně. První funkce načítá obrazy z kamer, které slouží ke kalibraci a případným nastavením před testováním setřením okenního skla. Druhá funkce slouží k načtení obrazu, který se provede po kontrolním setření skla. Tento obraz se používá ve funkcích, které ho poté zpracovávají.

K načítání obrazu ze souboru slouží *imaqReadFile*. Jedná se o funkci z knihovny *NI Vision*, která má tři parametry: cílový soubor, načítaný soubor a barevná hloubka. Tuto funkci využívaly 3 jiné mnou vytvořené funkce. Načtený obraz se následně zobrazoval v uživatelském okně. Chybné, či špatné načtení je v tomto okně indikováno pomocí informačních diod.

V uživatelském okně se obraz zobrazoval do *Canvasu*, pomocí funkce *Zobraz* (obr. 3.23). Jejím parametrem je číslo pole, ve kterém je zobrazovaný obraz uložen. Obraz je ve formátu *Image*, ale *Canvas* pracuje pouze s bitmapou, proto je nejdřív obraz

zkopírován do pole a poté převeden do bitmapy. Až poté je zobrazen pomocí *CanvasDrawBitmap* do bitmapy. Opět se jedná u funkci z knihovny *NI Vision*. Před ukončením této funkce se provede vymazání dočasných dat z paměti počítače.

```
//----- Zobrazí obraz do Canvasu -----//
int zobraz(int z)
{
    int PIX_DEPTH = 8;
    int i;

    for (i=0; i<256;i++) colorTable[i] = i*i*256+i*256*256;
    if(imageArray[z] == 0)
    {
        return -1;
        break;
    }
    imArray = imaqImageToArray (imageArray[z], IMAQ_NO_RECT, &bmp_width, &bmp_height);
    if(z == 2*NUM_IMAGES+1)
    {
        for (i=0; i<bmp_height*bmp_width;i++)if(imArray[i] > 0)imArray[i] = 255;
    }
    NewBitmap (bmp_width*PIX_DEPTH/8, PIX_DEPTH, bmp_width, bmp_height, colorTable, imArray, NULL, &bmpID);
    CanvasDrawBitmap (panelHandle, PANEL_CANVAS, bmpID, VAL_ENTIRE_OBJECT, VAL_ENTIRE_OBJECT);

    DiscardBitmap (bmpID);
    imaqDispose (imArray);
    return 0;
}
```

Obr. 3.23: Funkce zobraz

Dále jsou v souboru *hks.c* funkce zpracovávající a testující obraz. Některé tyto funkce jsou uvedeny níže.

```
//----- Spusteni difference mezi prvnim a druhym obrazem -----//
int CVICALLBACK DiferenceCallback (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            imageArray[2*NUM_IMAGES] = imaqCreateImage(IMAQ_IMAGE_U8, 0);
            imaqSubtract(imageArray[2*NUM_IMAGES], imageArray[0], imageArray[NUM_IMAGES + 0]);
            imageArray[2*NUM_IMAGES+1] = imaqCreateImage(IMAQ_IMAGE_U8, 2);
            imaqThreshold(imageArray[2*NUM_IMAGES+1], imageArray[2*NUM_IMAGES], 20, 255, TRUE, 1);
            imaqSetWindowPalette (3, IMAQ_PALETTE_BINARY, NULL, IMAQ_PALETTE_USER);
            imaqMoveWindow(3, MakePoint (280,400));
            zobraz(2*NUM_IMAGES+1);
            break;
    }
    return 0;
}
```

Obr. 3.24: Funkce Diference

Funkce difference (obr. 3.24) je spuštěna kliknutím na tlačítko difference v okně spuštěného programu. Nejdůležitější je zde funkce *imaqSubtract*, která má tři vstupní parametry. První z nich je výstupní obraz, druhým je obraz, který se odečte od obrazu, který je třetím parametrem. Odečítají se jasové hodnoty každého bodu a v případě že někde v obraze došlo ke změně jasu, tento rozdíl bude uložen jako jasová hodnota do výstupního obrazu na příslušnou pozici. Výsledný obraz se poté ještě prahuje, aby byly rozdíly více patrné. Prahování se provede funkcí *imaqThreshold*, která 6 vstupních parametrů: výstupní obraz, vstupní obraz, dolní mez prahu, horní mez prahu,

TRUE/FALSE hodnotu nastavující přepsání hodnot jasu v rozmezí na hodnotu která je posledním parametrem. Práh je možno manuálně upravovat v okně uživatelského rozhraní.

```
//----- Zjisteni prum hodnoty oblasti a dispose -----//
histogramMask = imaqCreateImage (IMAQ_IMAGE_U8, 2);
imaqROIToMask(histogramMask, region[POC_REG - 1], 1, NULL, NULL);
quantifyRpt_Ptr = imaqQuantify(imageArray[NUM_IMAGES + n], histogramMask);
GetCtrlVal(panelHandle, PANEL_Odchylka, &odch);
odchylka = odch + 0.0;
min = quantifyRpt_Ptr->global.mean - odchylka;
max = quantifyRpt_Ptr->global.mean + odchylka;
if(min < 0)min = 0;
if(max > 255)max = 255;
imaqDispose (userROI);
imaqDispose (histogramMask);
imaqDispose (quantifyRpt_Ptr);

imaqSetROIColor(region[POC_REG - 1], &a);
//----- Vytvoreni jednotlivych oblasti a jejich vykresleni pokud se lisi o "odchylku" -----//
for(i=0; i<(pocetk-1); i++)
{
    region[POC_REG - 1] = imaqCreateROI ();

    annulus.center.x = x;
    annulus.center.y = y;
    annulus.innerRadius = poc + (krok * i);
    annulus.outerRadius = poc + (krok * (i + 1));
    annulus.startAngle = s;
    annulus.endAngle = e;

    imaqAddAnnulusContour (region[POC_REG - 1], annulus);

    histogramMask = imaqCreateImage (IMAQ_IMAGE_U8, 2);
    imaqROIToMask(histogramMask, region[POC_REG - 1], 1, NULL, NULL);
    quantifyRpt_Ptr = imaqQuantify(imageArray[NUM_IMAGES + n], histogramMask);

    if(quantifyRpt_Ptr->global.mean < min || quantifyRpt_Ptr->global.mean > max)
    {
        imaqOverlayROI (imageArray[NUM_IMAGES + n], region[POC_REG - 1], IMAQ_POINT_AS_CROSS, NULL, NULL);

        point = imaqMakePoint(x, (y + (krok / 2) + (krok * (i + 1))));
        sprintf ("pom", "%d", i + 1);
        imaqOverlayText(imageArray[NUM_IMAGES + n], point, "ROI - Region zajmu", &a, &overlayTextOptions, NULL);
    }

    imaqDispose (userROI);
    imaqDispose (histogramMask);
    imaqDispose (quantifyRpt_Ptr);
}
```

Obr. 3.25: Funkce Prum

Funkce průměrování již byla částečně popsána v kapitole 3.3.1.3. Nejprve se označí ROI, ve kterém se bude hledat chyba. Tato oblast se označí za pomoci dvou prstenců vytvořenou funkcí *NastavROI*, ve které se mimo deklarace proměnných a nastavení okna výběru oblasti, nachází taktéž funkce z knihovny *NI Vision*, *imaqConstructROI*. Tato funkce má 6 parametrů: zobrazený obraz, ROI, strukturu (*IMAQ_ANNULUS_TOOL*), nastavení okna (*&toolWindowOptions*), nastavní ROI (*&ROIOptions*) a kontrolní proměnnou. Slouží k vytvoření ROI. K překreslení ROI do obrazu se používá funkce *imaqOverlayROI*. Jedná se o funkci z knihovny *NI Vision*, která má 5 parametrů: obraz na který se nakreslí ROI, ROI, a symbol, který reprezentuje překreslující bod, hodnota bodů a skupina ke které chcete vrstvu přidat. Poté se z načtených hodnot vypočítá střed, začátek a konec celkové oblasti a šířky jednotlivých oblastí, na které je celková oblast rozdělena. Tato funkce skončí a po ní se spustí napsaná funkce *Prum*, jejíž část je vidět na obrázku 3.25. Zde se vypočítává průměrná

hodnota jasu oblasti, která se poté porovnává s jednotlivými částmi oblasti, na které je rozdělena. Pokud je rozdíl hodnoty jasu nějaké z těchto oblastí větší než odchylka, je tato oblast uložena do příslušného pole oblastí a poté zakreslena a označena.

Dále jsou zde obsaženy také pomocné funkce, například k zadávání vytvoření ROI pomocí bodů, či vyhledání středu kružnic při jejím zadání třemi body. Konkrétně tyto funkce slouží k zadání různých tvarů ROI. Paleta zadávaných tvarů v knihovně *NI Vision* je omezena a proto se speciální tvary musí naprogramovat.

```
int Bod (int i)
{
    int okay0;
    ContourID pointID;
    ContourInfo2 *pointInfo = NULL;
    int contourCount;
    ToolWindowOptions toolWindowOptions;
    ConstructROIOptions ROIOptions;

    toolWindowOptions.showClosedFreehandTool = FALSE;
    toolWindowOptions.showFreehandTool = FALSE;
    toolWindowOptions.showLineTool = TRUE;
    toolWindowOptions.showOvalTool = TRUE;
    toolWindowOptions.showPointTool = TRUE;
    toolWindowOptions.showPolygonTool = TRUE;
    toolWindowOptions.showPolyLineTool = TRUE;
    toolWindowOptions.showRectangleTool = FALSE;
    toolWindowOptions.showRotatedRectangleTool = FALSE;
    toolWindowOptions.showZoomTool = TRUE;
    toolWindowOptions.showAnnulusTool = TRUE;
    toolWindowOptions.showPanTool = FALSE;
    toolWindowOptions.showSelectionTool = FALSE;
    imaqSetWindowSize(NULL, 1500, 1000);

    ROIOptions.type = IMAQ_PALETTE_GRAY;
    ROIOptions.palette = NULL;
    ROIOptions.windowNumber = 0;
    ROIOptions.windowTitle = "ROI";

    imaqConstructROI(imageArray[NUM_IMAGES + i], region[POC_REG - 1], IMAQ_POINT_TOOL, &toolWindowOptions, &ROIOptions, &okay0);

    contourCount = imaqGetContourCount (region[POC_REG - 1]);
    pointID = imaqGetContour (region[POC_REG - 1], 0);
    pointInfo = imaqGetContourInfo2 (region[POC_REG - 1], pointID);
    point = *(pointInfo->structure.point);
    imaqDispose(pointInfo);
    return 0;
}

Point skruznice (Point a1, Point a2, Point a3)
{
    double k1, k2, q1, q2;
    Point s;
    //vyp smernice kolme k primce
    k1 = (double)(a2.y - a1.y) / (double)(a2.x - a1.x) * (-1.0);
    k2 = (double)(a3.y - a2.y) / (double)(a3.x - a2.x) * (-1.0);
    if(k1 == k2) printf("rovnobezky");
    q1 = ((a1.y + a2.y) / 2.0) - (k1 * ((a1.x + a2.x) / 2.0));
    q2 = ((a2.y + a3.y) / 2.0) - (k2 * ((a2.x + a3.x) / 2.0));
```

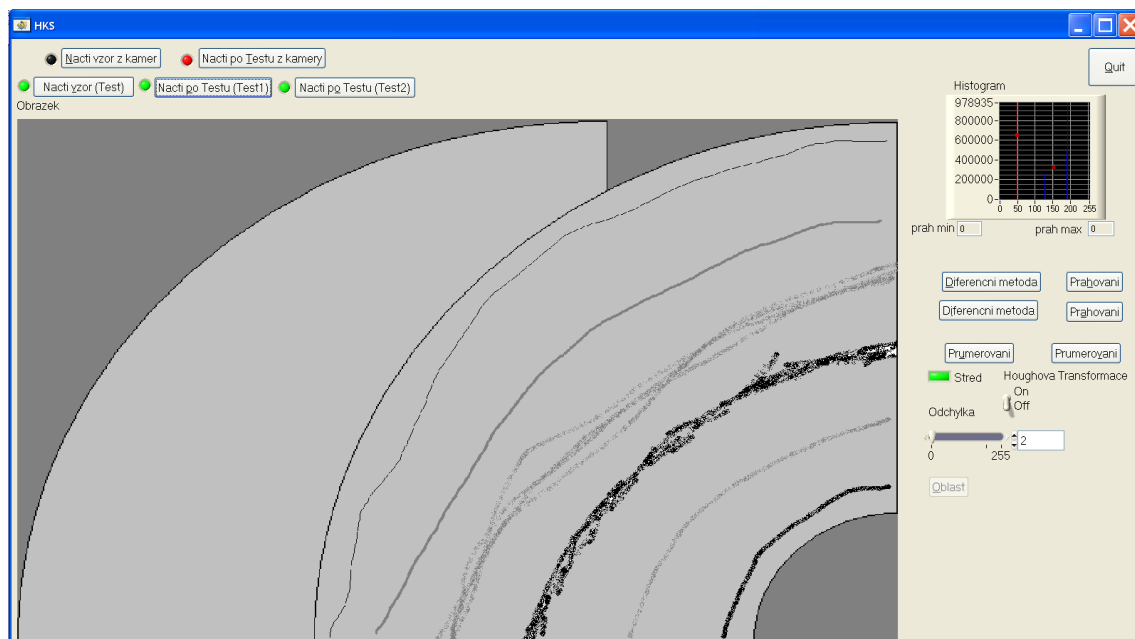
Obr. 3.26: Funkce pomocné

3.3.2.2 Grafické uživatelské rozhraní

Okno uživatelského rozhraní (obr. 3.27) obsahuje různá tlačítka, posuvníky, přepínače, diody a plochu ve které se zobrazuje obraz (Canvas).

V levé horní části se nachází 5 tlačítek pro načítání obrazu. Z toho dvě jsou pro načítání obrazu z kamer a 3 pro načítání ze souboru. U každého tlačítka je informační

dioda, která se rozsvítí červeně v případě chyby, zeleně v případě, že načtení proběhlo v pořádku a černá zůstane pokud tlačítko nebylo doposud použito.



Obr. 3.27: Okno grafického uživatelského rozhraní programu

Pravá část okna, která slouží k ovládání programu, obsahuje tlačítka a posuvníky, histogram a informační diodu. Dokud nejsou splněny některé podmínky (například načtení obrazu), nelze tlačítka použít, vyjma tlačítka *Quit*, které slouží k ukončení programu a lze ho použít vždy. Toto tlačítko se nachází v pravém horním rohu a hned pod ním se nachází histogram. Ten se zobrazí až po načtení obrazu. Poté lze červenými posuvníky měnit hranici prahu a upravit si ji dle potřeb. Spodní a horní hranice je zobrazena pod histogramem, níže se nachází tlačítka *Diference* a *Prahování*, která slouží ke spuštění stejnojmenných funkcí. Jejich výsledek je zobrazen v Canvasu, nebo novém okně. Pod těmito tlačítky se nachází *Prumerování*. Po kliknutí na toto tlačítko se otevře nové okno, kde se zadá pomocí vnitřní a vnější okraj oblasti, ve které se bude průměrovat (viz obr. 12 v příloze). Šířkou oblasti je průměrná šířka okrajových oblastí, které byly zadány. Střed obou dvou krajních oblastí zadávaných uživatelem se musí nacházet přibližně na stejném místě, jinak se okno zavře, průměrování neprovede a informační dioda u tlačítka se rozsvítí červeně. V případě, že vše proběhlo v pořádku, se tato dioda rozsvítí zeleně. Po provedení funkce se otevře okno s oblastmi, které obsahovaly průměrný jas o rozdílu větším než je difference (viz obr. 12 v příloze). Ta se nastavuje pod tlačítkem *Prumerování*. V těchto místech lze pomocí přepínače nastavit

používání Houghovy transformace, která se provede po průměrování. Nejnižší je tlačítko oblast, kterým se nastaví ROI o specifických tvarech, které se poté vymaskují.

Zbylou, tedy největší část okna zabírá Canvas, do kterého se vykresluje obraz v podobě načtených dat, výsledků, či mezivýsledků.

3.4 Zhodnocení NI Vision Development Module

Vision Development Module nemá prakticky žádného konkurenta jelikož není nikdo, kdo by nabízel podobně obsáhlý a komplexní programový balík pro oblast zpracování obrazu a počítačového vidění. Je vyvinut a uzpůsoben pro práci s obrazem, a proto je v něm práce s obrazovými daty nesrovnatelně lepší a jednodušší, než je tomu například v některém editoru jazyka C, či Matlabu, které nejsou tak jednostranně profilovány. Také díky tomu obsahuje velké množství funkcí pro práci s obrazem, z nichž jsou některé alogirtmy a funkce patentovány. S nimi se dá pracovat ve více než jednom programovacím editoru, které tento programový balík obsahuje, a to díky tomu, že se zaměřuje na širokou masu lidí a některým skupinám více vyhovuje jiný styl programování. V případě potřeby svým zákazníkům firma National Instruments, poradí, popřípadě i doporučí potřebný hardware, jelikož spolupracuje s velkým množstvím firem z oblasti počítačového vidění a informatiky.

Ani tento programový balík není všemocný nástroj k práci s obrazem, který by uměl vše, a proto je čas od času potřeba něco naprogramovat. Ale i přesto obsahují knihovny velké množství funkcí se kterými se dobře pracuje a díky tomu je programování i složitějších úloh jednodušší a rychlejší než u klasických programovacích jazyků, které nejsou takto profilovány na zpracování obrazu. Funkce jsou optimalizované, jak pro rychlost, tak i pro přesnost. Své algoritmy a funkce si firma National Instruments chrání patenty. V programovém balíku je obsažen Vision assistant, který slouží především k otestování a předzpracování. Co která funkce, nebo množina funkcí dělá je ihned vidět na testovaném obrázku po kliknutí myši. Programový balík obsahuje rozsáhlou nápovědu, příklady a dokumentaci, což přijde vhod především začátečníkům. V této nápovědě je z počátku složitější orientace, ale po tom, co si uživatel zvykne na to, co a kde hledat bývá nápověda dobrým pomocníkem.

4 Závěr

Zpracování obrazu se využívá mnoha v odvětvích a profesích, například ke zvýšení efektivity, přesnosti, snížení nákladů, kontrolám, či výzkumu a v průmyslu je využíváno již od sedmdesátých let minulého století. K masovějšímu využití došlo až počátkem tohoto století, k čemuž mimo jiné přispělo snížení cen a zkvalitnění čipů v kamerách. Díky nadále klesajícím cenám zájem neustále roste, proto je jedním z výsledků diplomové práce popis nejrozsáhlejšího a nejkomplexnějšího programového balíku určeného ke zpracování obrazu, kterým je NI Vision Development Module. Popsány byly především funkce a knihovny pro zpracování obrazu a vývojová prostředí NI Vision Assistant, NI LabVIEW a NI LabWindows/CVI.

Na praktické úloze bylo předvedeno, co vše je potřeba k získání obrazových dat do počítače, a že se jedná o velmi důležitý postup, který nám může následné zpracování obrazu ulehčit, ztížit, nebo dokonce úplně znemožnit. Dle zadání se musely zvolit vhodné kamery, objektivy a osvětlení, tak aby z načteného obrazu bylo možné co nejjednodušeji dosáhnout potřebných výsledků. Někdy je taktéž potřeba použít speciálních zařízení, jako tomu bylo u osvětlení pomocí elektro luminiscenčních panelů při této úloze.

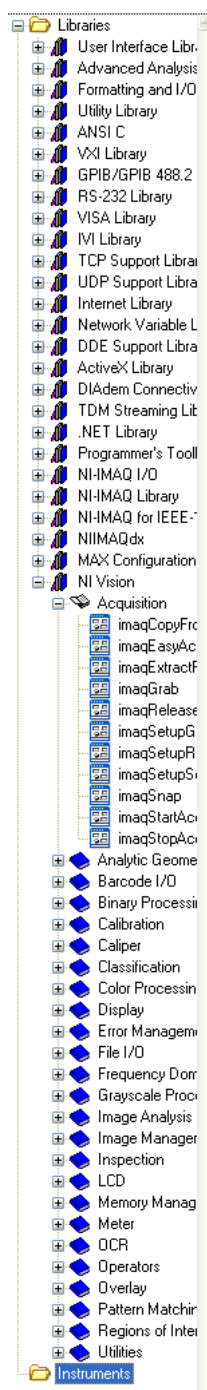
K vytvoření programu posloužil jak NI Vision Assistant, tak především NI LabWindows/CVI. Nejprve byly promyšleny vhodné metody, které jsou v této práci taktéž popsány, a poté následovalo testování a zkoušení různých jednoduchých metod a jejich různých nastavení v NI Vision Assistant. Samotný program a grafické uživatelské rozhraní bylo vytvořeno v LabWindows/CVI, kde byla využívána především knihovna NI Vision, která obsahuje velké množství funkcí, především pro práci s obrazem.

Na konec je možné říci, že i přes různé potíže, které zapříčinila především špatná kvalita předního skla, která znemožnila použití některých funkcí, se podařilo nalézt vhodnou metodu. Jedná se o diferenční metodu, která si s těmito problémy poradila a zbývající vodu po setření na skle spolehlivě odhalovala.

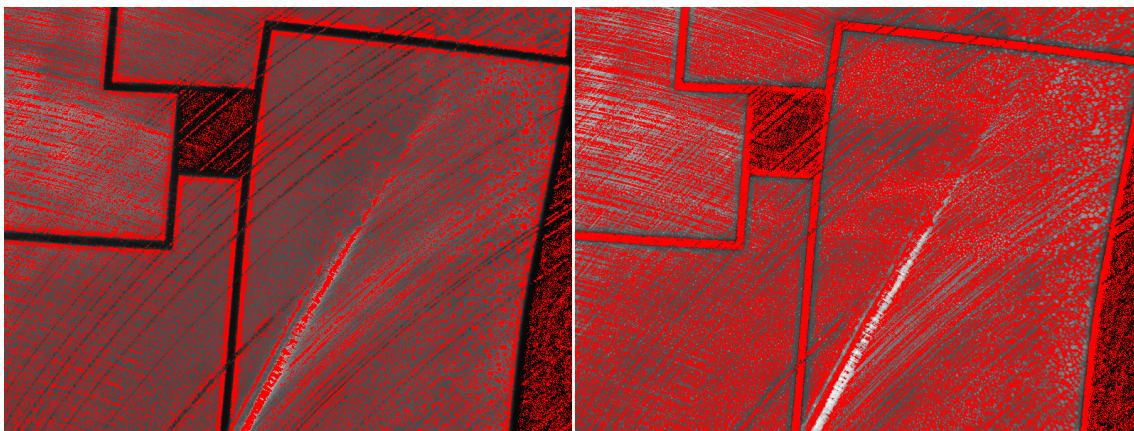
Seznam literatury

- [1]HLAVÁČ, V., SEDLÁČEK, M. *Zpracování signálů a obrazů*. Praha : Skriptum ČVUT, 2002.
- [2]SONKA, HLAVAC, BOYLE *Image Processing Analysis and Machine Vision*. PSW publishing, 2002.
- [3]NI Vision Concept Manual, National Instruments 2005.
- [4]*Www.ni.com* [online]. 2007 [cit. 2009-04-06]. Dostupný z WWW: <www.ni.com>.

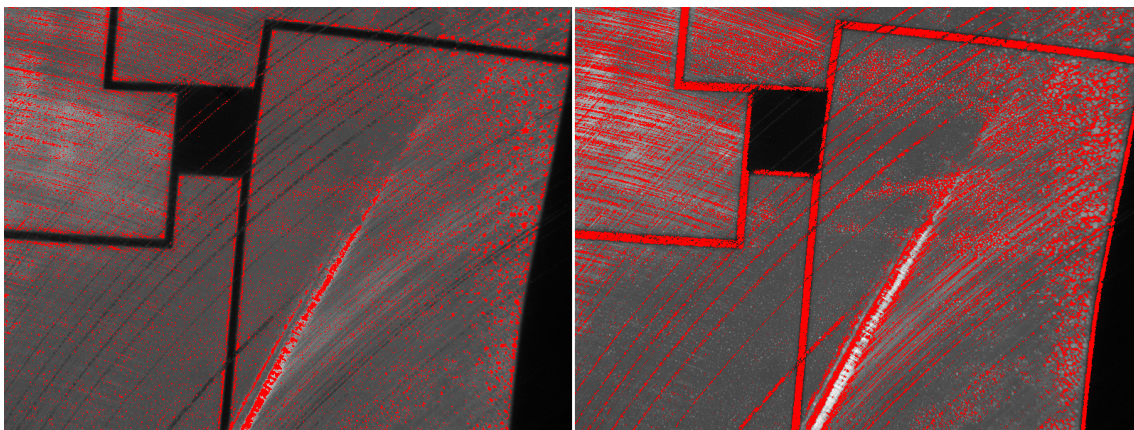
Přílohy:



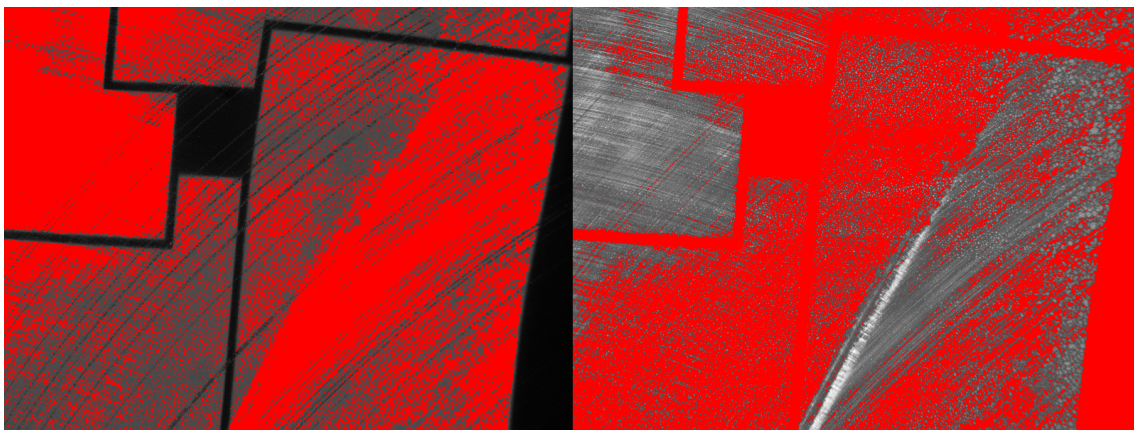
Obr. 1: Lokální prahování, Niblack (světlé objekty, tmavé objekty)



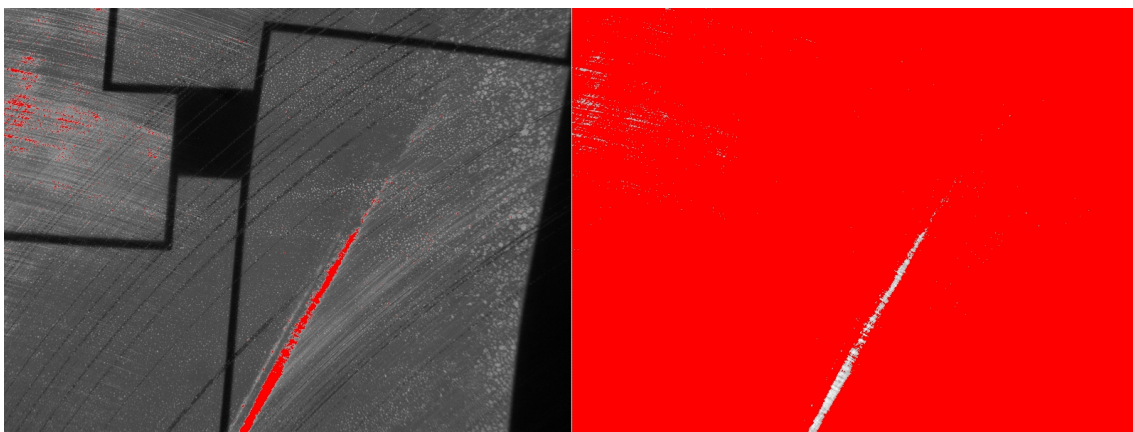
Obr. 2: Lokální prahování, Niblack (světlé objekty, tmavé objekty)



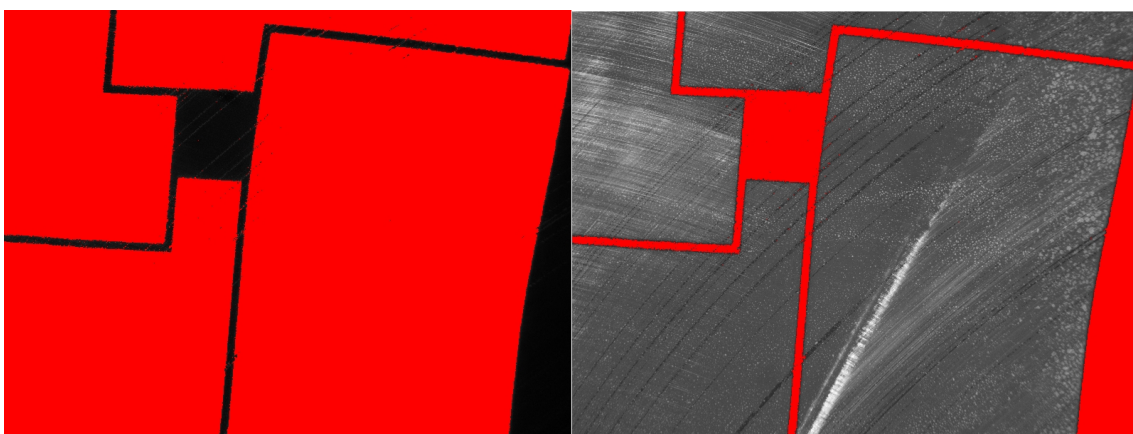
Obr. 3: Lokální prahování, Background Correction (světlé objekty, tmavé objekty)



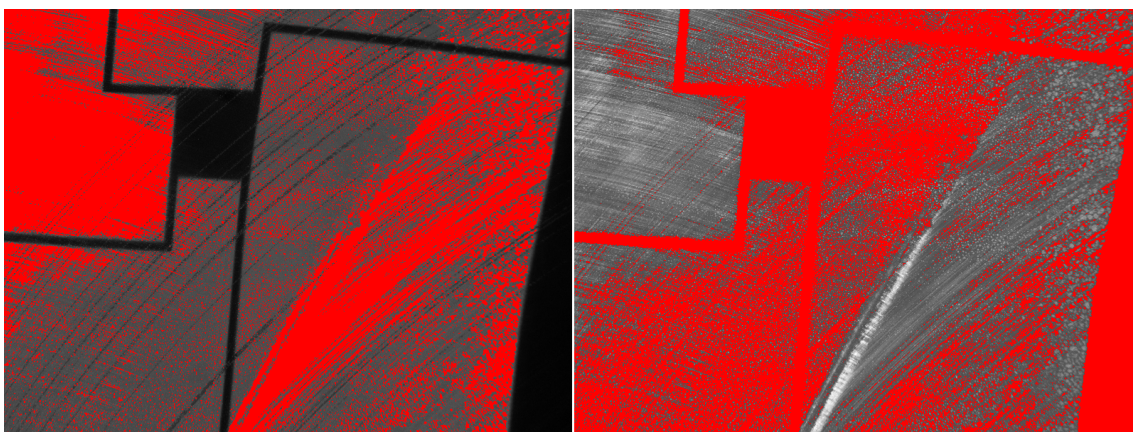
Obr. 4: Automatické prahování, Clustering (světlé objekty, tmavé objekty)



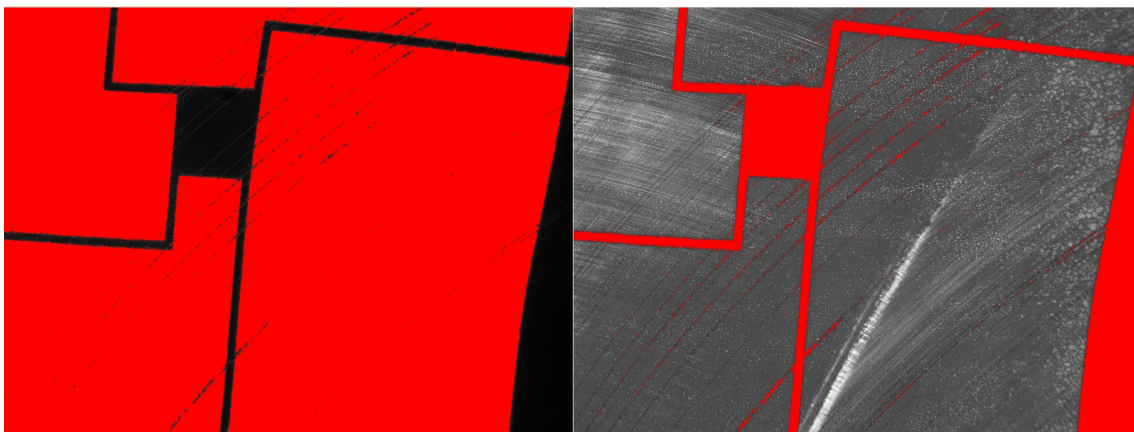
Obr. 5: Automatické prahování, Entropy (světlé objekty, tmavé objekty)



Obr. 6: Automatické prahování, Metric (světlé objekty, tmavé objekty)



Obr. 7: Moments, Clustering (světlé objekty, tmavé objekty)


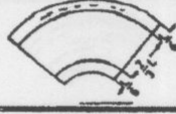









Obr. 8: Inter Variance, Clustering (světlé objekty, tmavé objekty)

8.5 Wischqualität statisch

Siehe Tabelle 2.










Tabelle 2 – Wischqualität an einer Prüfscheibe

Wischfehler	Note	Definition			Fehler
		Bereich	Zustand	Anzahl	
	10	-	-	-	keine
	9	1/6 L	nicht bleibend	1	<i>jetzt provisch</i> Nadelstreifen $\leq 0,3 \text{ mm}$
	8	1/6 L	nicht bleibend	max. 3	Nadelstreifen $\leq 0,3 \text{ mm}$
		1/6 L	bleibend	1	Nadelstreifen $\leq 0,3 \text{ mm}$
	7	1/6 L	bleibend	max. 3	Nadelstreifen $\leq 0,3 \text{ mm}$
		2/3 L	nicht bleibend	1	Nadelstreifen $\leq 0,3 \text{ mm}$
	6	1/6 L	bleibend	max. 4	Streifen $\leq 1,0 \text{ mm}$
		1/6 L und 2/3 L	nicht bleibend	max. 10	Nadelstreifen $\leq 0,3 \text{ mm}$
		2/3 L	bleibend	1	Nadelstreifen $\leq 0,3 \text{ mm}$

*L. die
ramen*

Obr. 9: Strana 19 z normy TL 934

D

Wischfehler	Note	Definition			Fehler
		Bereich	Zustand	Anzahl	
	3	1/6 L	bleibend <i>trivial</i>	max. 6	Streifen ≤ 1 mm
		1/6 L und 2/3 L	nicht bleibend	max. 16	Nadelstreifen ≤ 0,3 mm
		2/3 L	bleibend	max. 2	Nadelstreifen ≤ 0,3 mm
	4	1/6 L und 2/3 L	bleibend	Viele	Streifen unter- schiedlicher Breite, jedoch 75% der über- strichenen Fläche muß gewischt werden.
	3	1/6 L und 2/3 L	bleibend	Viele	Streifen wie bei 4, jedoch zentraler im Bereich 2/3 L bzw. überstrichene Fläche ≤ 75 %
	2	1/6 L und 2/3 L	bleibend	-	Wasserinseln
		1/6 L und 2/3 L	bleibend	-	Krallendurchdrücken
		1/6 L und 2/3 L	bleibend	-	starker Nachschleier
	1	1/6 L und 2/3 L	bleibend	-	Rattermarken

Obr. 10: Strana 20 z normy TL 934

8.6 Wischqualität dynamisch

Siehe Bild 2 und Tabelle 3.

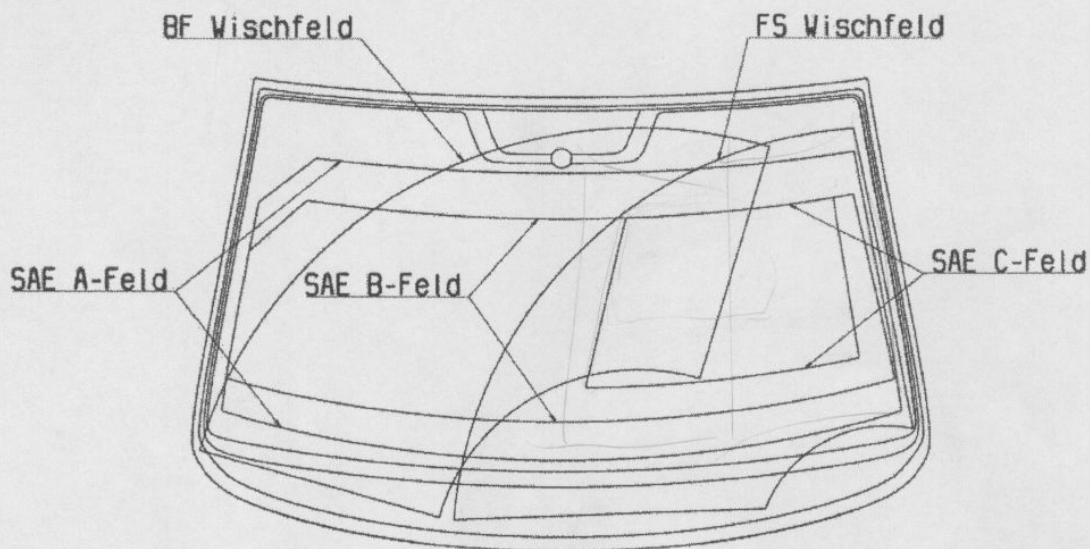


Bild 2

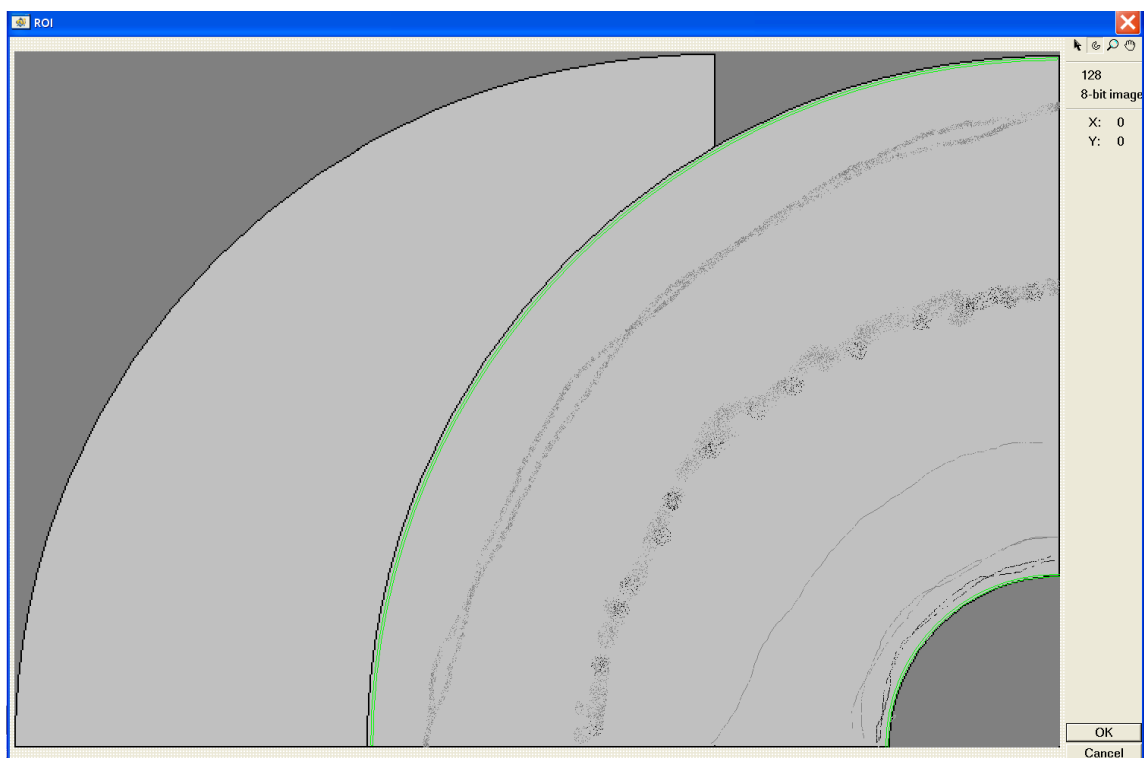
Tabelle 3

Geschwindigkeit bis	SAE A-Feld	SAE B-Feld	SAE C-Feld	Regensensorfläche
170 km/h	x	-	-	-
180 km/h	x	-	-	-
190 km/h	x	-	-	-
200 km/h	x	x	-	-
210 km/h	x	x	-	-
220 km/h	x	x	O	-
230 km/h	x	x	O	-
240 km/h	x	x	O	-
250 km/h	x	x	O	-
X = zulässige Wischfehler/Wassernachziehen				
O = keine vollflächigen Wischfehler bei Geschwindigkeiten über 220 km/h				

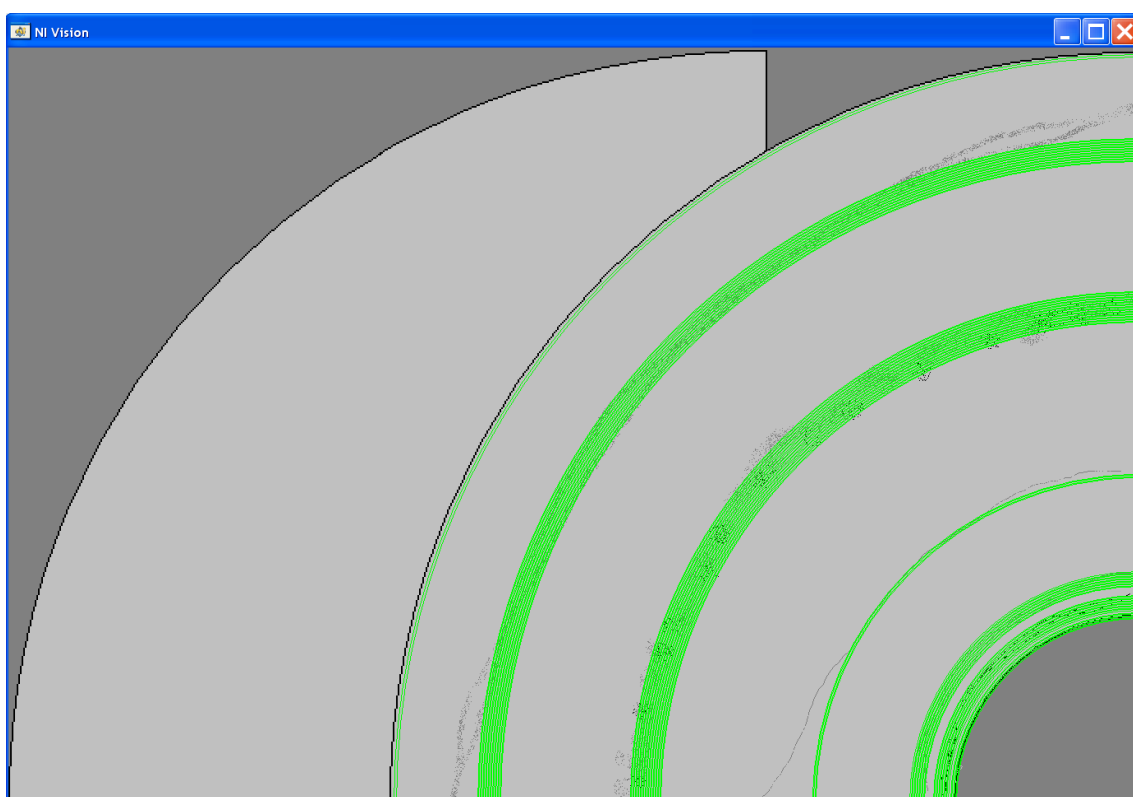
9 Mitgeltende Unterlagen

VW 01155	Fahrzeug-Zulieferteile; Genehmigung von Erstlieferung und Änderung
VW 50185	Fahrzeugbauteile; Freibewitterungsbeständigkeit
VW 80101	Elektrische und elektronische Baugruppen in Kraftfahrzeugen
VW 91101	Umweltnorm Fahrzeug, Fahrzeugteile, Werkstoffe, Betriebsstoffe; Schadstoffvermeidung

Obr. 11: Strana 21 z normy TL 934



Obr. 12: Vytváření ROI u funkce průměrování



Obr. 13: Nalezené ROI s chybou větší než odchylka